# Development of a verifiably accurate road lighting calculation application in accordance with EN13021-3:2003

John Everist

MSc Light and Lighting 2006

**CONTENTS**

## 0. ABSTRACT

Road lighting installations are required to meet performance figures determined by a region's governing body in order to assure that the scheme will provide adequate lighting for road users after dark. For many countries in Europe including the UK, road lighting standards are dictated by the European Committee for Standardisation (CEN). To establish whether a proposed lighting installation will meet its relevant targets, a method of prediction must be used to ascertain the light distribution over the considered area. This is achieved by using mathematical formulas to calculate the results of the proposed installation. Traditionally luminaire manufacturers would provide charts, tables and graphical tools derived form these formulas so that a conforming scheme could be established without having to perform complex calculations. Today however, these methods of light computation are considered obsolete by the majority of people in the industry as lighting software will generate and present results far quicker than manual processes.

The aim of this project was to create a system for verifying the accuracy of road lighting software in accordance with EN 13201-3. This has been addressed by developing an application within *Microsoft Excel* that follows these conventions exactly and also presents intermediate calculation stages in spreadsheets so that its methods of computation

can be understood. This paper details its development and also identifies issues found in the standard as a result of this project.

# 1. INTRODUCTION

The onset of darkness results in a huge reduction in the comfort and performance of the human visual system. A result of this, is that road users are considerably more likely to be involved in an accident after dark. Studies in America have shown that over 50% of their road deaths occur at night and statistics in the UK show that the occurrence of accidents after dark are 180% of that by day. Visual comfort and performance can be improved through the introduction of an artificial lighting installation and consequently the likelihood of an accident be decreased. Studies in many counties and by many institutions have shown that road lighting definitely contributes to a reduction in night-time accidents. In general, research shows a reduction of over 30% after a road lighting installation was provided. It has also been long established that the running cost of a road lighting installation can to a large extent be offset against the cost saving in preventing the accident[1].

A road lighting installation should provide an adequate amount of visual information for all road users to ensure that safe travel is not hindered by inadequate visual conditions. This can only be obtained if both visual performance and visual comfort are satisfactory. In order to insure that the visual needs of road users are met, countries and organisations produce standards of lighting which must be met. These standards are the products of the latest research and developments in technology, and therefore felt to provide the best lighting solution for the needs of road

[1] Bommel, W.J.M. and Boer, J.B. (1980)

users. An installation's suitability is evaluated primarily on the quantity of light obtained on the ground, the uniformity of this light and the amount of glare received from the installed luminaries.

When calculating the quantity of light provided by an installation two units are considered being *luminance* and *illuminance*. On main roads where traffic is predominately vehicular the visual needs of drivers are most important and so the measure of light is luminance; this is because a driver needs to be able identify small objects at low contrasts on the road surface. On minor roads where traffic is slower moving, the visual needs of non motorised traffic such as bicycles or pedestrians must be considered. As the area of interest is much closer and panoramic the principles of luminance calculations can not be applied and so the measure of light in these situations is illuminance.

The distribution of light achieved over the considered surface is referred to as *overall uniformity* and can affect a driver's ability to adequately see at all locations on the road. This is because light from the road surface surrounding dark areas interferes with the image of that area. Uniformity is measured as the ratio between the average and minimum values and the smaller the ratio the worst the eye's ability to detect objects in the lowest lit areas of the road. Where light is being measured in luminance an additional calculation of the uniformity directly in front of drivers is required as a continuous sequence of bright and dark patches on the road in front of a driver will impact on visual comfort and therefore visual performance. This effect is referred to a *longitudal uniformity* and is defined as the ratio between the minimum and maximum road surface luminance on a line parallel to the road axis running through the observer position.

Whilst glare can take either of two forms being *discomfort glare* and *disability glare*, it is only disability glare that is referred to when evaluating a road lighting installation. An object that can just be seen in the absence of glare (threshold contrast) can not be seen when glare is present unless the actual contrast is increased. This effect forms the basis of the measure for loss of visual performance due to glare and is referred to as *Threshold Increment (TI)*. TI is defined as the amount of extra contrast required to again just make the object under glare conditions, divided by the effective contrast. The value of TI is influenced by the light distribution from the lantern between 70 and 90 degrees in elevation, the surface brightness of the road, the layout of the lanterns, the mounting height and the observer angle[2],[3].

The illuminance achieved at the roadside is also important, as it is a method of ensuring that there is adequate visibility for pedestrians and drivers to detect people and objects to the side of the carriageway. The lighting of a strip along side the road should be bright enough so that this is possible, but not so bright as to change the adaptation level of the driver's vision and as a result reduce the visibility of the road. This

[2] Bommel, W.J.M. and Boer, J.B. (1980)
[3] Pritchard, D. C. (1999)

requirement is referred to as the *surround ratio* and is defined as the ratio between the average illuminance of a 5 metre strip either side of the road to a 5 metre strip in the road at the edge of the carriageway. In areas where pedestrians are excluded it is unnecessary to consider the surround ratio[4].

The requirements that road lighting installation must meet in order to provide adequate visual conditions are dependant upon the intensity, speed, composition of the traffic and complexity of the road system. Many European countries including the UK base their road lighting standards on those formulated by CEN. The requirements for a road lighting design in the UK are set out in two standards; BS 5489 – 1:2003 and BS EN 13201. BS 5489 – 1:2003 specifies the lighting classes set out in EN 13201-2 and gives guide lines on the application of these classes by including a system to define traffic areas in terms of parameters relevant to lighting. EN 13201 has three sections which relate to the various processes the designer of on installation will need to consider. EN 13201-2 defines classes for road lighting according to the photometric requirements of the needs of road users and also considers environmental aspects of road lighting. It uses *ME classes* to define the requirements for main roads and *S classes* to define the requirements for minor roads. EN 13201–3 defines the calculation methods to be used to establish the characteristics of an installation so that results obtained from different sources will have a uniform basis. EN 13201-4 establishes

[4] Raynham, P. (2006)

the conventions and procedures for lighting measurements of road lighting installation and gives advice on the use and selection of luminance meters and illuminance meters[5],[6].

The CIE has also published technical report *CIE 140-2000* which dictates the methods of calculation for road lighting. This report is actually a revision of a previous publication *CIE 30.2*. It includes the calculation procedures for luminance and illuminance, and introduces modified practices for the location of calculation grid points and observer positions. The conventions stated in CIE140-2000 are the same to that of EN 13201-3. CIE 30.2 included a listing of two computer programs for carrying out calculations which could be used as a benchmark to assess the accuracy of other programs, these have been omitted from the current revision due to difficulties maintaining the them. The programs which supported CIE 30.2 where referred to as *Standard Luminance Program (STAN)* and *Comprehensive Luminance Program (LUCIE)*; STAN was a simple program capable of calculating straight sections of road whist LUCIE was able to simulate more complicated scenarios. The age of the programs now mean that they are incompatible with current operating systems and are therefore inaccessible to the vast majority of people[7],[8].

---

[5] *CEN/TR 13201-1*. BSI (2004)
[6] *EN 13201-2, EN 13201-3, EN 13201-4.* (2004)
[7] *CIE 140-2000.* (2000)
[8] *CIE 30-2.* (1982)

In order to perform calculations information describing the photometric qualities of the proposed luminaire and road surface are required. The photometric qualities of road surfaces are described through reflection tables referred to as *R-tables*; these tables can be used generate a reduced co-efficient dependant on an assumed viewing angle of 1 degree below horizontal, the lights angle of incidence onto the road surface and the angle of deviation from its original path to that of the observer. R-tables for a range of road surface types are defined in CIE 66 (1984), in the UK however, R-Table C2 is typical for the majority of road surfaces and calculation tools issued by manufacturers are based on this.

The intensities of luminaries are given in tables are referred to as *I-tables*, these state intensities achieved at measured angles about the luminaire. This data can be used in mathematical formulas to calculate the illuminance and, when used in conjunction with an R-table, luminance. Calculating illuminance or luminance in this manner for an entire installation is a slow and repetitive task. As a result manufactures supply calculation tools which can show the performance of the luminaire numerically or graphically in a far more efficient manner. The most common calculation aid is the *performance sheet* shown in figure 1.1 as it shows the achieved illuminances and luminances for different layouts; this means that the impact of any modification can be quickly explored. Specific to illuminance schemes *utilisation factor curves* can be used to quickly establish overall luminaire quantities and on more complicated road layouts *isolux diagrams* can be used to define column locations. Similar tools are available for schemes measured in luminance and are referred to as *Luminance Yield Diagrams* and *Isoluminance Diagrams*[9].

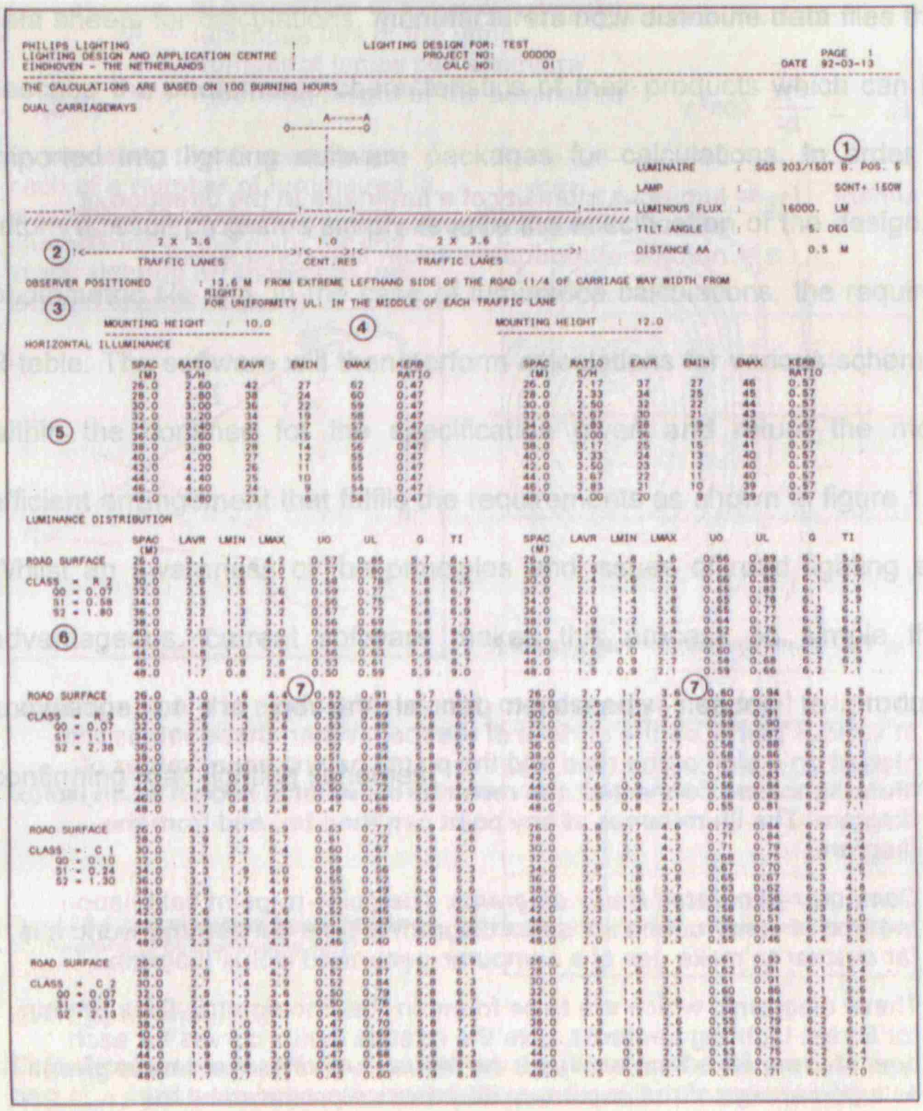



Fig. 7.26 A typical lighting scheme, or performance sheet, showing the data contained in it:
1. Type of luminaire, light source, and general installation data
2. Cross-section of the road showing the lighting arrangement
3. Specification of observer position
4. The two mounting heights for which the sheet is valid
5. Illuminance values
6. Class of road surface
7. Luminance, uniformity and glare values

---

[9] *Philips Lighting Manual 5th Edition.* Philip Lighting B.V. (1993)

***Figure 1.1***

*Sample luminaire performance sheet.*

Computer software is now the main method of calculating the photometric qualities of a road lighting installation. Rather than provide data sheets for calculations, manufacturers now distribute data files that describe the photometric characteristics of their products which can be imported into lighting software packages for calculations. In order to return a result programs simply require the specification of the design, a photometric file and, in the case of luminance calculations, the required R-table. The software will then perform calculations for various schemes within the confines for the specification given and return the most efficient arrangement that fulfils the requirements as shown in figure 1.2. Whilst an awareness of the principles and issues of road lighting are advantageous, current software makes this process so simple that knowledge of the industry is not necessarily needed to produce confirming road lighting schemes.

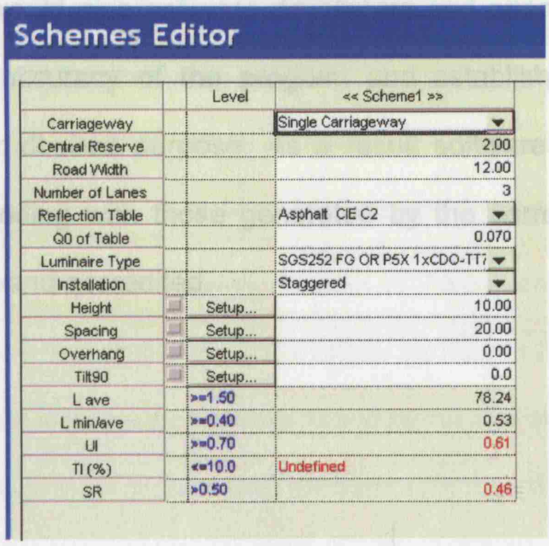



***Figure 1.2***

*Specification and result page of Calculux.*

The simplicity of the interface of software means that whilst values are returned very quickly, there is no supporting data to show how the result was achieved. This means that users are assuming that the software is using the correct mathematical procedures and the appropriate conventions to generate the results. If a program differs from the stated methods then the results it generates may not meet the desired standards and as a consequence, the installed scheme might supply inadequate lighting for road users. It is not realistic or useful to the vast majority of people to be able to view the algorithms of the programs which generate the results. However, a different and viable method of inspection would be to compare the results of the software in question to that of those generated by a verified system. A program, which could generate the same setting as the popular software currently available, would provide a method of like for like comparison for all achievable schemes. This would give software developers and operators a means to analyse the accuracy of the program and establish whether it is suitable for their desired purpose. As a result software which returns values corresponding with those generated by the correct procedures can be identified and accredited.

## 2. PROJECT OBJECTIVES

The specifications of virtually all road lighting schemes designed and installed at present are defined by a computer program's optimisation and calculation routines. The transition from manual processes to computer methods is predominantly a result of the accelerated workflow and reduced need for employee knowledge the use software offers a company. Computer generated results are now an industry standard as it is assumed that their results are precise; it is common practice to include a computer generated report in client presentations as proof that a proposal will be compliant with standards. However, whilst EN13201-3 defines the mathematical formulas and conventions which should be used when calculating road lighting schemes CEN does not offer any tools or procedures to verify the accuracy of software.

The goal of this project is to generate a method by which the accuracy of software in relation to the calculation procedures stated in EN13201-3 can be measured. Whilst using the precise conventions to generate results for example installations would provide a means of comparison, this solution would have a limited usefulness as it would be restricted to a finite number of variations. In order to provide a truly versatile system that would enable like for like comparisons for any scheme, a device needs to be developed that can deal with the same variables as the road lighting software currently available. This is only achievable by developing a road lighting application which implements the procedures in EN13201-3 precisely and has the same functionality as other road lighting software. The primary focus of such an approach is to implement a means by which the accuracy of the program can be verified else the irony of checking one unverified program against another will render the application useless.

The simplest way to prove the program is using the correct conventions is to make the calculation process as transparent as possible. This requires all the intermediate processes and results be shown in addition to the absolute values returned at the end. This way a user can follow the procedures the application is using and if desired, verify they are in accordance with EN 13201-3. Such a method will quickly generate a large volume of numerical data and so a method of logging and presenting this data meaningfully is required; this will be achieved using Microsoft's spreadsheet application *Excel.* A spreadsheet is particularly suited to this task its primary function is to process and display large amounts of numerical data and has all the required mathematical functions and presentation tools inbuilt. Working with a pre-made platform also alleviates the needs to design and build an application from start. Whilst other spreadsheets such as *Lotus 123* are capable of the same functions, Excel is a stronger candidate as it is installed on the majority of computers and therefore an idea way of distributing the application.

The basic structure of a spreadsheet is to have cells containing values which are the products of mathematical procedures defined in part by the contents of other cells. This concept can be expanded into many layers until a complex matrix is achieved capable of processing data in the required manner. This is not an absolutely ideal approach as firstly, entering data into cells can be an awkward and confusing process to users not familiar with Excel and secondly, if absolutely every process was displayed the quantity of data might be overwhelming. Excel allows for customised procedures, functions and interfaces to be created using *Visual Basic for Applications (VBA)* meaning that the application within Excel can be tailored to best fulfil its requirements. Using VBA, very low level procedures can be processed but not displayed so that the remaining data presented is done so more clearly and is therefore easier to understand. Users could also enter data into the application via specifically created forms with labels, input sections and error handlers to minimise confusion and user mistakes.

It is imperative that all lighting applications be able to import photometric data from manufacturers, without this function software would only be able to calculate with preloaded information chosen by the developers. The need for a method to circulate photometry has resulted in a number of standardised file formats which all applications can import and translate into intensity values. Whilst the official photometric format for Europe is the CEN file, this format is unused and the most popular format used is the *elumdat file (.LDT, .ELX, .LUM);* this is unofficially the European standard format. The application must therefore have the functionality to browse a hard-drive, load and assimilate an elumdat file in order that the necessary intensities can be derived. R-tables are far fewer in number and rarely need to be modified, a result of this absence of demand means that a standardised format has never been developed. Whilst the most commonly used R-tables could be pre-loaded into the application, this would be a laborious process and again limit the functionality of the software. A better solution would be to enable the application to load and assimilate the R-tables of another program; as the file format of an R-table is similar in structure to that of elumdat files this will be a relatively simple function to develop.

Finally, whilst calculation transparency will enable developers and operator to personally verify that the software is giving appropriate results, this system is dependant on people having the mathematical ability to analyse the processes and access to EN 13201-3. A solution to this issue is to involve respected members of the lighting community to assist in the development of the software by checking the procedures and commenting on the program during development. This would give reassurance of the program's accuracy to those unable or uninterested in verifying it personally. The figures invited to be involved in this process have been chosen from varying backgrounds in an effort to eliminate any type of bias and diversify input. The involved are Peter Raynham who lecturers in lighting at University College London; Jan Koster who is head programmer for Philips Lighting; and Ron Simons who wrote EN 13201-3.

## 3. METHODOLOGY

The application operates by using the VBA procedures and maths functions of Excel to create a workbook containing several worksheets which display the calculated data. The worksheets used in the application are: *Results Summary* which displays summarised data as required by EN 31021-2; *Results Details* which contains accumulated data for all the luminaires in the project; *Luminaire1, Luminaire2 etc* which shows the calculations for each luminaire in isolation; *Photometry* and *R-table* which displays the I-table and R-table loaded; and finally a hidden worksheet called *Data* to hold filepaths. A sample session is shown in figure 3.0.1. The workflow of the project development was aimed to follow the chronological stages a user would interact with the application as this enabled completed sections of the program to be sent to the checking bodies for verification in manageable sections; this also enabled any errors to be identified and corrected before additional formulas where generated from these figures. The development order this method dictated was creating user-forms for data entry, reading and interpolating photometric data, setting up the scheme components, calculating illuminance, luminance and glare, and finally generating quality figures. Errors where recognized and dealt in a far more random order but they have been identified and described in this structure in order to maintain continuity in this report.

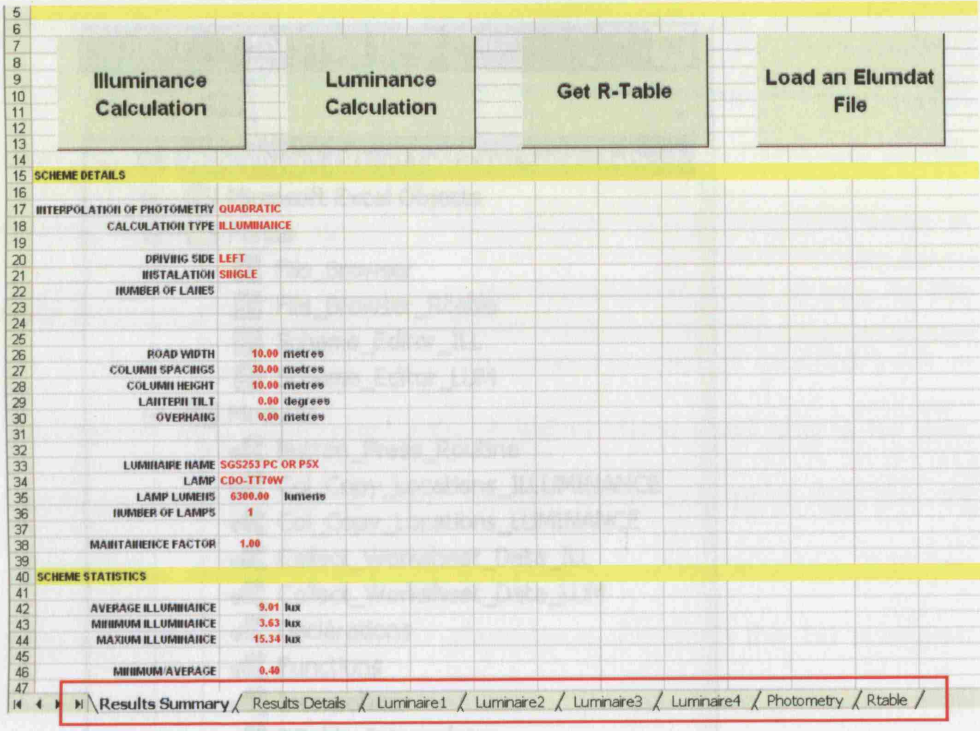



*Figure 3.0.1*

*Results Summary worksheet of a sample project*

The Visual Basic code in the application was divided into userforms and modules relevant to the tasks it performed so that the code was broken into more manageable sections; in retrospect other methods could have achieved this without the associated complications of so many modules. The list of modules and userforms of the project are shown in figure 3.0.2.

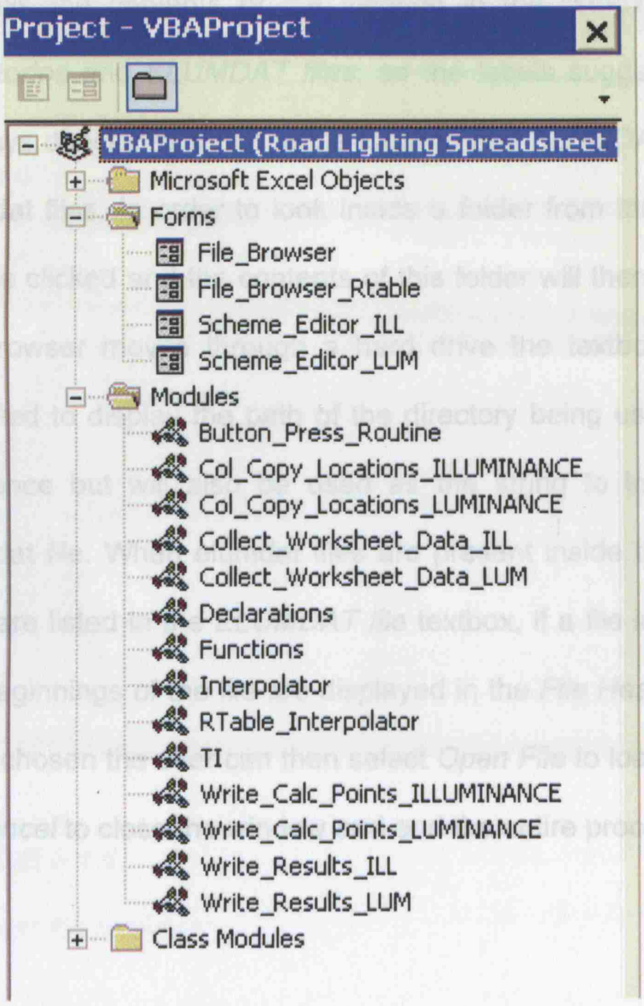



***Figure 3.0.2***

*List of the modules and forms as shown in VBA*

## 3.1 FILE BROWSER

As Excel does not have a pre-made file browsing system for VBA projects, a browser was created using a userform called *File_Browser* comprising *textboxes* and *listboxes* to accommodate all the requirements of a browser; the form is shown in figure 3.1.1. The top window labelled drive is a textbox and requires the user to enter the name of the required drive in order to start browsing. When a valid drive is entered the form displays the contents of the location in the listboxes below labelled *Directories* and *ELUMDAT files*; as the labels suggest *directories* only displays directories found in the location and *ELUMDAT files* only shows elumdat files. In order to look inside a folder from this list, the folder is double clicked and the contents of this folder will then be shown. As the file browser moves through a hard drive the textbox labelled *path* is modified to display the path of the directory being used, this is for user reference but will also be used as the string to load in the desired elumdat file. When elumdat files are present inside the selected folder, they are listed in the *ELUMDAT file* textbox, if a file in this list is clicked the beginnings of the file are displayed in the *File Heading* window. With a file chosen the user can then select *Open File* to load it into the project or *Cancel* to close the window and end the entire process.

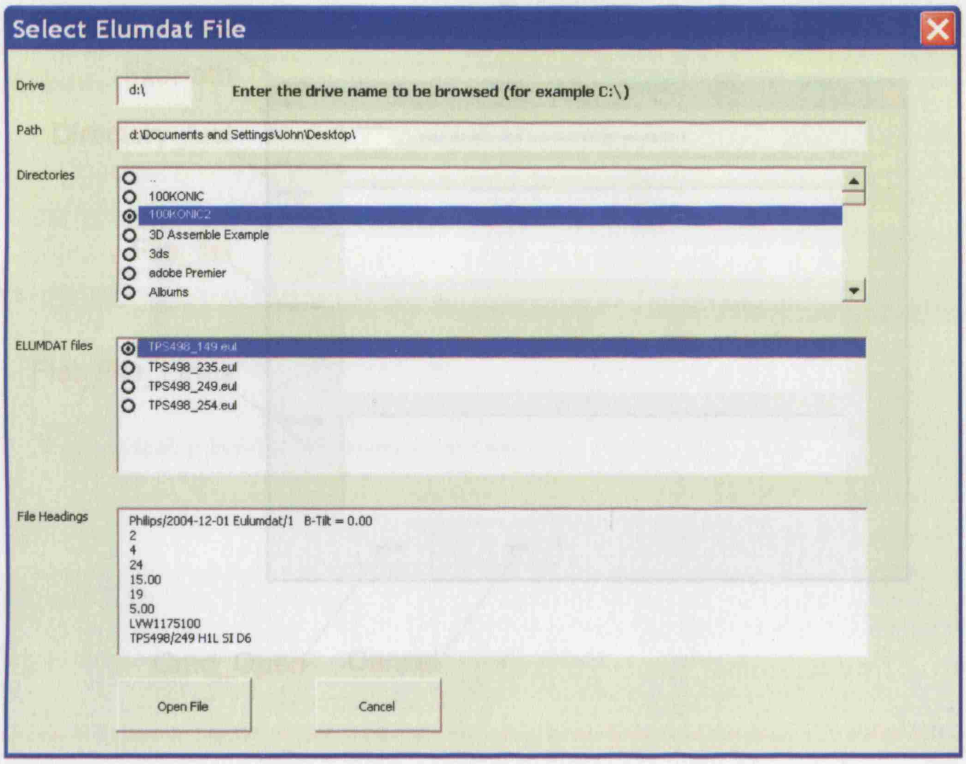



***FIGURE 3.1.1***

*Elumdat file browser*

The values held in the forms are accessed in VBA using the names which they were allocated during their creation; the names used in this form are shown in figure 3.1.2.

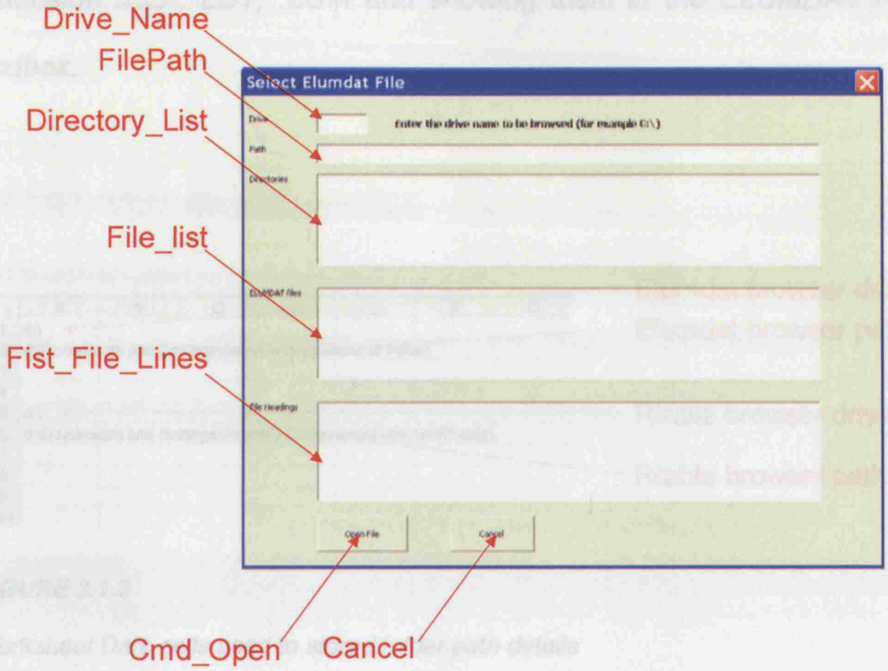



*FIGURE 3.1.2*
*Component names of the elumdat file browser*

These values are treated as *strings* in VBA and the browser actually functions by using routines which analyse and modify them in order to look in different folders. The drive name and file path are held in cells in a hidden worksheet called *Data*, this means that the browser locations will be remembered after the window is closed (even if Excel is closed); these cells are shown in figure 3.1.3. When a drive reference is typed into *Drive_Name* the value is first written into the *Data* worksheet and then the routine *Update_Dir* is triggered which using the *Dir()* command and the *Check_If_Directory function* all folders in this location are listed in the *Directories* listbox. The routine *Update_Files* is then triggered which repeats this process only looking for elumdat files with the extension *.ELX, LDT, .LUM* and showing them in the *ELUMDAT Files textbox.*

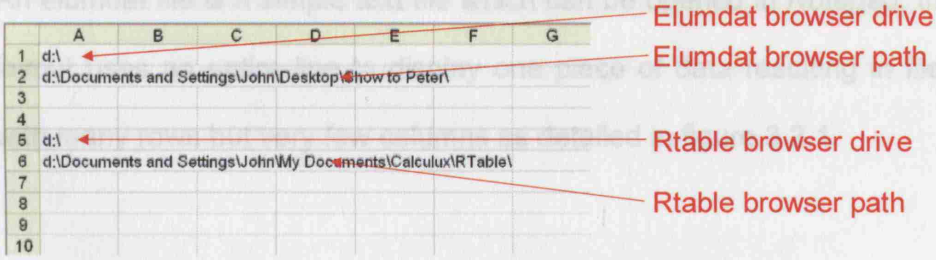


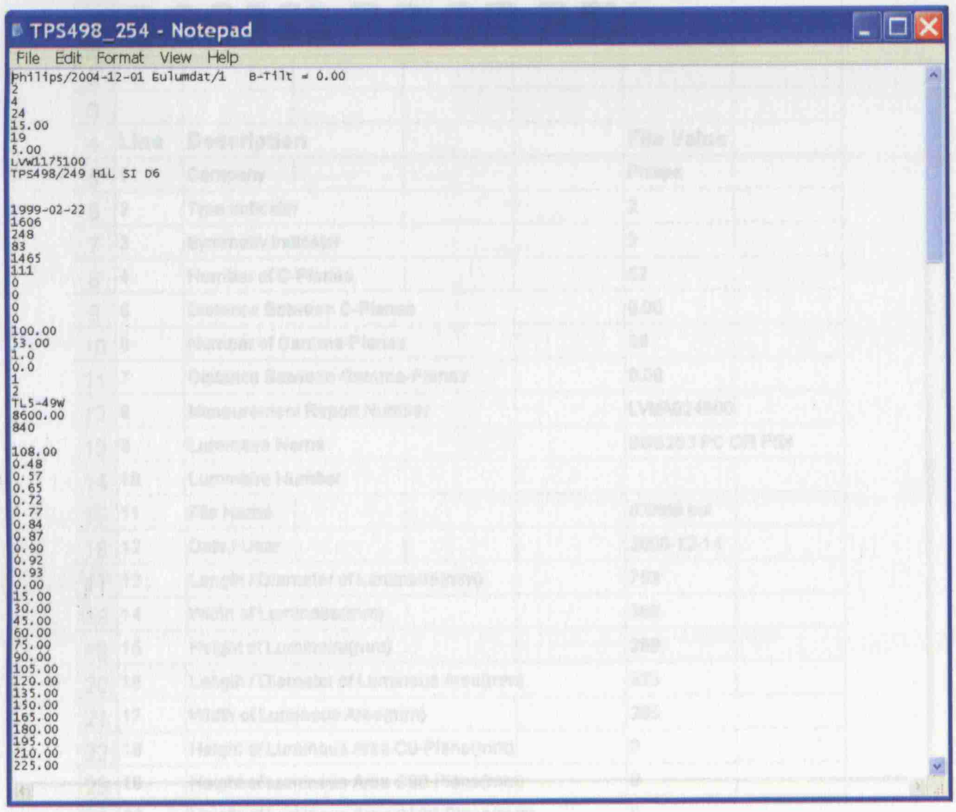

***FIGURE 3.1.3***

*Worksheet Data cells used to stored folder path details*

When a directory is double clicked from the *Directories* list the routine *Directory_List_DblClick* is triggered, this adds the name of the directory selected to that of the existing path and writes it into the *Data* worksheet thus creating a new path name; this path name is displayed in the *filepath* textbox. The *Update_Dir* and *Update_File* routines are then repeated using the new path details. Clicking ".." uses the *Left()* command at the top of the directory list to remove the last directory from the file and browse the parent directory. When an item in the file list is selected the routine *File_List_Click* is triggered which using the *Open For Input As* command writes the beginnings of the file into the *File Heading* window. Clicking the *File Load* button will begin importing the file whilst the *Cancel* button will hide the form and exit the routine. The code for the *File_Browser* userform is contained in the form itself and sample code showing the *Update_Dir* function is listed in appendix A (page 85).

## 3.2 FILE READER AND WRITER

An elumdat file is a simple text file which can be opened in *Notepad*, the layout uses an entire line to display one piece of data resulting in files with many rows but very few columns as detailed in figure 3.2.1.



```
TPS498_254 - Notepad
File  Edit  Format  View  Help
Philips/2004-12-01 Eulumdat/1   B-Tilt = 0.00
2
4
24
15.00
19
5.00
LVW1175100
TPS498/249 H1L SI D6


1999-02-22
1606
248
83
1465
111
0
0
0
0
100.00
53.00
1.0
0.0
1
2
TL5-49W
8600.00
840

108.00
0.48
0.57
0.65
0.72
0.77
0.84
0.87
0.90
0.92
0.93
0.00
15.00
30.00
45.00
60.00
75.00
90.00
105.00
120.00
135.00
150.00
165.00
180.00
195.00
210.00
225.00
```

***FIGURE 3.2.1***

*Sample elumdat file opened in Notepad*

The data order of the file is to first list general luminaire information, then the angles measured and finally the intensities recorded; this structure is used as the number of angles and intensities will fluctuate depending on the number of angles measured. Using the *filepath* string the application opens the file and uses the *Line Input* command to write the general data into the *Photometry* worksheet re-creating the file with data titles as shown in figure 3.2.2.

| | A | B | F |
|---|---|---|---|
| 1 | **SGS253 PC OR P5X** | | |
| 2 | | | |
| 3 | | | |
| 4 | **Line** | **Description** | **File Value** |
| 5 | 1 | Company | Philips |
| 6 | 2 | Type Indicator | 2 |
| 7 | 3 | Symmetry Indicator | 3 |
| 8 | 4 | Number of C-Planes | 52 |
| 9 | 5 | Distance Between C-Planes | 0.00 |
| 10 | 6 | Number of Gamma-Planes | 38 |
| 11 | 7 | Distance Between Gamma-Planes | 0.00 |
| 12 | 8 | Measurement Report Number | LVMA024900 |
| 13 | 9 | Luminaire Name | SGS253 PC OR P5X |
| 14 | 10 | Luminaire Number | |
| 15 | 11 | File Name | d:\lrid0.eul |
| 16 | 12 | Date / User | 2000-12-14 |
| 17 | 13 | Length / Diameter of Luminaire(mm) | 753 |
| 18 | 14 | Width of Luminaire(mm) | 368 |
| 19 | 15 | Height of Luminaire(mm) | 289 |
| 20 | 16 | Length / Diameter of Luminous Area(mm) | 335 |
| 21 | 17 | Width of Luminous Area(mm) | 295 |
| 22 | 18 | Height of Luminous Area C0-Plane(mm) | 0 |
| 23 | 19 | Height of Luminous Area C90-Plane(mm) | 0 |
| 24 | 20 | Height of Luminous Area C180-Plane(mm) | 0 |
| 25 | 21 | Height of Luminous Area C270-Plane(mm) | 0 |
| 26 | 22 | Downward Flux Fraction DFF(%) | 99.00 |
| 27 | 23 | Light Output Ratio Luminaire(%) | 79.00 |
| 28 | 24 | Conversion Factor for Luminous Intensities | 1.00 |
| 29 | 25 | Tilt of Luminaire During Measurement | 0.00 |
| 30 | 26 | Number of Standard Sets of Lamps | 1 |
| 31 | 27 | Number of Lamps | 1 |
| 32 | 28 | Type of Lamps | CDO-TT70W |

Results Summary / Results Details / Luminaire1 / Luminair

*FIGURE 3.2.2*

*Initial data of an elumdat file displayed in the Photometry Worksheet*

As the number of measured angles and intensities of an elumdat file will vary for different files, the I-table structure is communicated to software using numerical values which describe its symmetry and measured intensities; this data is stated at the beginning of the file as shown in figure 3.2.3.

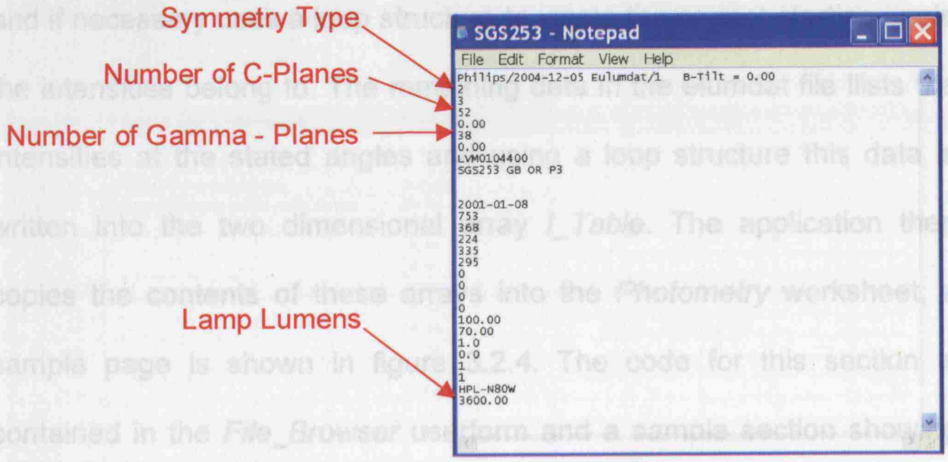



*FIGURE 3.2.3*

*Beginnings of an elumdat file with data crucial to road lighting identified*

These values are written into the variables *Sym_Typ*, *Num_Gamma_Angles* and *Num_C_Angles* when the general luminaire data is read. The next data the application encounters is a list of the measured C-planes and Gamma angles, using the values in *Num_Gamma_Angles* and *Num_C_Angles* the application redimensions the arrays *C_Angles* and *Gamma_Angles* to the appropriate size and writes in the angles in. Dependant on the luminaire's symmetry, it is possible that the stated number of angles is larger than the number of stated intensities and so these arrays need to be adjusted. The variable *Sym_Type* holds the symmetry type value which informs the routine of the number of C-planes available (a value for *type indicator* is not required as luminaires are always considered point sources in road lighting calculations). Based on the value of *Sym_Type* the routine modifies *Num_C_Angles* to match that of the actual quantity of angles and if necessary uses a loop structure to locate the correct starting angle the intensities belong to. The remaining data in the elumdat file llists the intensities at the stated angles and using a loop structure this data is written into the two dimensional array *I_Table*. The application then copies the contents of these arrays into the *Photometry* worksheet, a sample page is shown in figure 3.2.4. The code for this section is contained in the *File_Browser* userform and a sample section showing how the symmetry type value is used to determine which C-planes have been photometered is listed in appendix B (page 88)

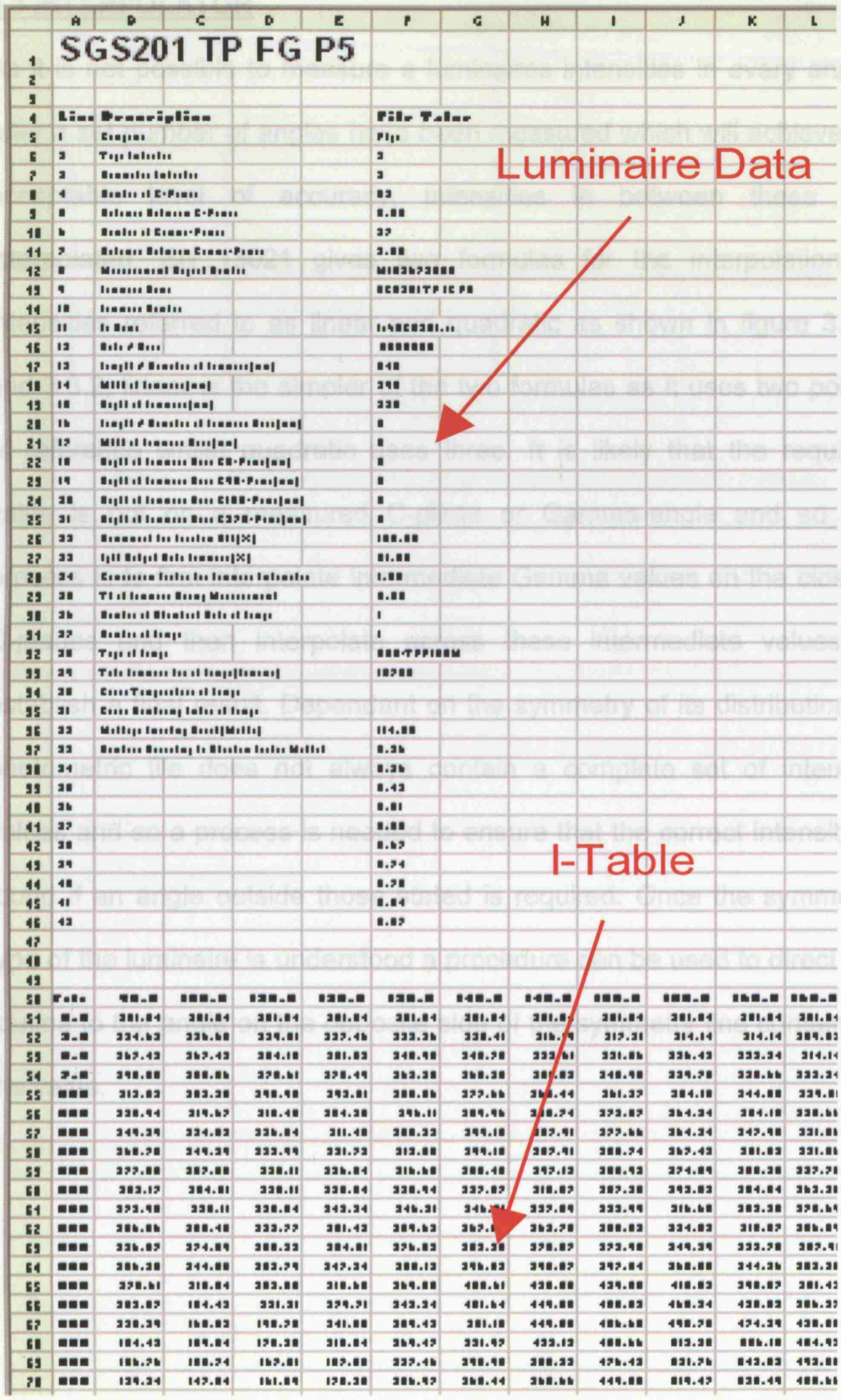



***FIGURE 3.2.4***

*Photometry worksheet with elumdat file loaded in.*

## 3.3 INTERPOLATOR

As it is not possible to measure a luminaires intensities in every angle, after a set number of angles have been measured which will achieve an acceptable level of accuracy, intensities in between these are interpolated. EN 13021 gives two formulas for the interpolation of intensities referred to as linear and quadratic as shown in figure 3.3.1 and 3.3.2; linear is the simpler of the two formulas as it uses two points of reference whilst quadratic uses three. It is likely that the required value is not on a measured C-plane or Gamma-angle and so the process is to first interpolate intermediate Gamma values on the closest C-planes and then interpolate across these intermediate values to establish a final result. Dependant on the symmetry of its distribution, a photometric file does not always contain a complete set of intensity values and so a process is needed to ensure that the correct intensity is found if an angle outside those stated is required. Once the symmetry type of the luminaire is understood a procedure can be used to direct the routine to the angle on the opposite side of the symmetry line containing the value.

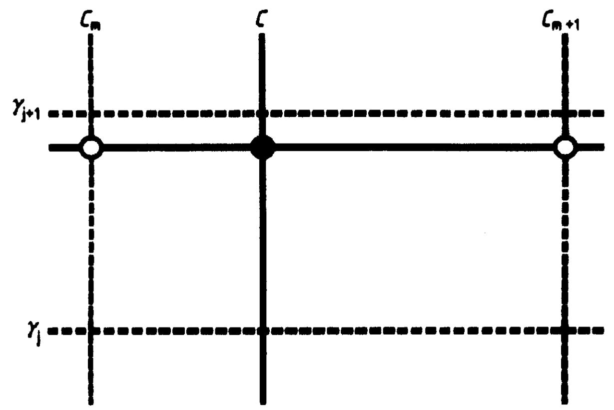



**Figure 2 — Angles required for linear interpolation of luminous intensity**

For this purpose, the following equations or mathematically equivalent equations shall be used:

$$K_1 = \frac{C_m - C}{C_m - C_{m+1}} \quad (3)$$

$$K_2 = \frac{\gamma_j - \gamma}{\gamma_j - \gamma_{j+1}} \quad (4)$$

***FIGURE 3.3.1***

*Formula for linear interpolation*[10]

[10] *EN 13201-3.* BSI (2004)

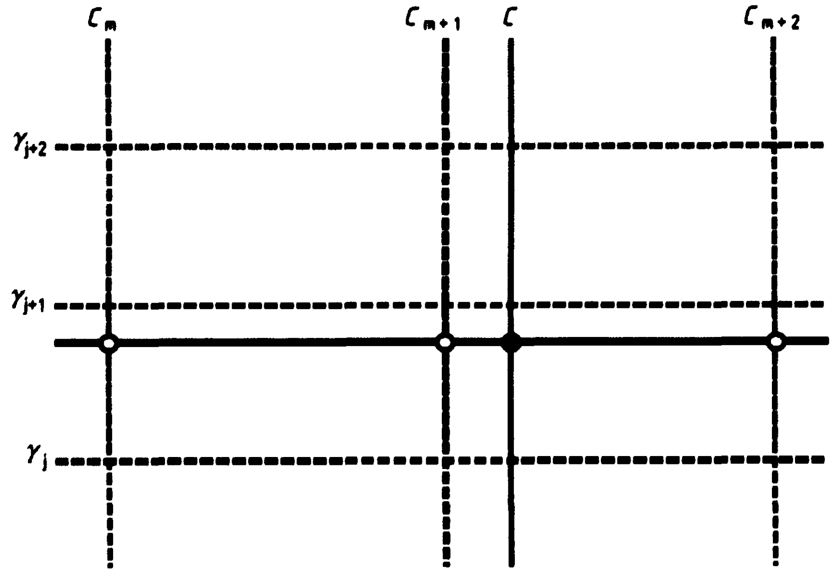



**Figure 3 — Values required for quadratic interpolation**

When interpolation is carried out in the region of $C = 0°$, or $\gamma = 0°$ or $180°$, see 5.3.4.

The formula for quadratic interpolation is

$$y(x) = y_1\left(\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}\right) + y_2\left(\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}\right) + y_3\left(\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}\right)$$

***FIGURE 3.3.2***

*Formula for quadratic interpolation*[11]

When an intensity is required the function *Quadratic_Inten* is called, this first triggers the routine *Read_Intensity* which this reads the I-table in the *Photometry* worksheet into a two-dimensional array called *Int_Values*; to optimise this process the routine checks to see whether the I-table has already been read to avoid repeating this task unnecessarily. Once the I-table has been read the function then calls the additional functions *Vert_Type* and *Hor_Type* to return a value that indicates the distribution and symmetry type of the luminaire. With these values the routine is then

[11] *EN 13201-3.* BSI (2004)

able to establish all the angles and intensities adjacent to the required point in any direction by calling the *Quadratic_Az_Point, Azimuth_Map, All_Azimuth, All_Elevation, Quadratic_El_Point, All_Intensity* functions. As quadratic interpolation requires three points of reference and linear interpolation requires only two, the same routine can be used to collect reference values for both and in the case linear interpolation the last value discarded as only two reference points are required. The convention for quadratic interpolation means that the required intensity could be located between either the first and second point or second and third; as the linear interpolation routine rejects the last value an error would occur if the adjacent points where second and third. Additional code was implemented to ensure that the third value was always the greatest of the three when the interpolation routine was linear.

The process of calculating linear interpolation is comparatively simple to that of quadratic and consequently the entire formula for linear interpolation is contained in the *Linear Interpolation* function. As the formula for quadratic is far more complex in order for the process to be easier to analyse the process has been spread across three routines being *Quadratic_I3, Quadratic_K1, Quadratic_K2,* and *Quadratic_K3*. *Quadratic_I3* is the initial function which returns the final value, the other three functions perform the intermediate processes. The code used to perform the interpolation routines is contained in the *Interpolator* module and both interpolation routines are listed in appendix C (page 91).

In order the verify the accuracy of the of interpolation routines values were first calculated for the measured angles across the C-planes and Gamma-angles on a sample road optic and are plotted onto charts for analysis as all three should clearly be identical; these charts are is shown in figures 3.3.3 and 3.3.4 whilst full size charts are shown in appendix D (page 93).

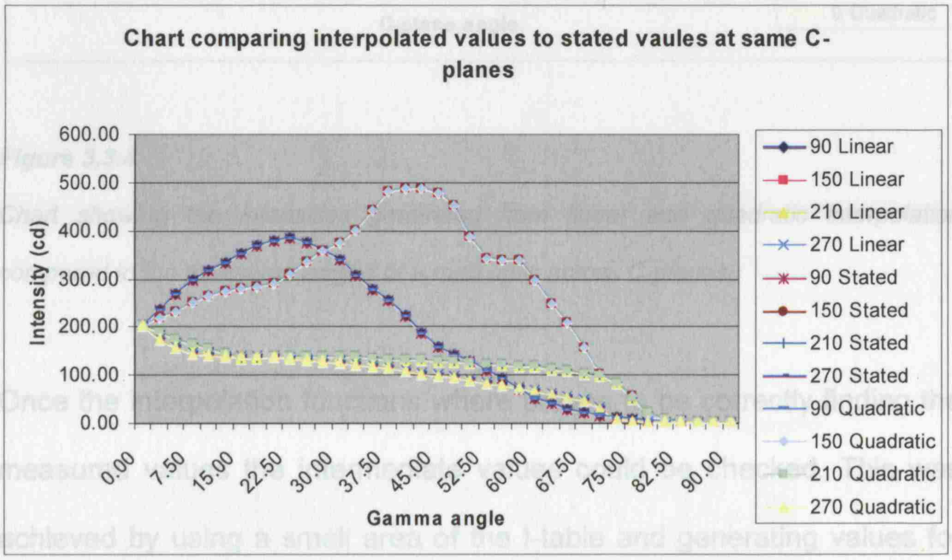



*Figure 3.3.3*

*Chart showing the intensities generated from linear and quadratic interpolation compared to the measured angles of a road optic across Gamma angles.*

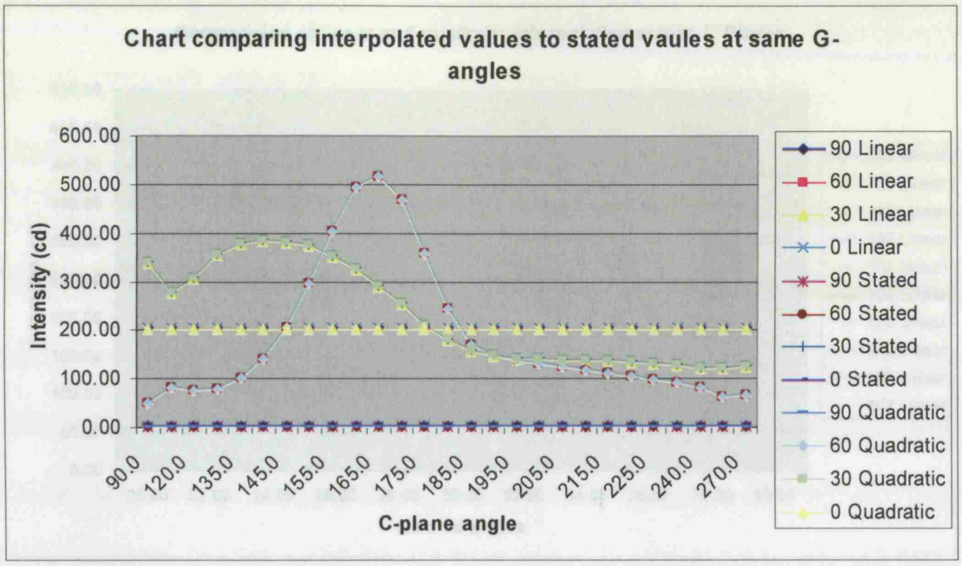



*Figure 3.3.4*

*Chart showing the intensities generated from linear and quadratic interpolation compared to the measured angles of a road optic across C-planes.*

Once the interpolation functions where proven to be correctly finding the measured values the intermediate values could be checked. This was achieved by using a small area of the I-table and generating values for both measured and non-measured angles and again plotted the data into a chart; both linear and quadratic where put onto the same chart as whilst theu use different routines their values should be very similar. The pattern of the chart could then be analysed as a correctly functioning interpolation routine should create a smooth pattern from one measured value to the next; these charts are shown in figures 3.3.5 and 3.3.6 with full size charts in appendix D (page 93).

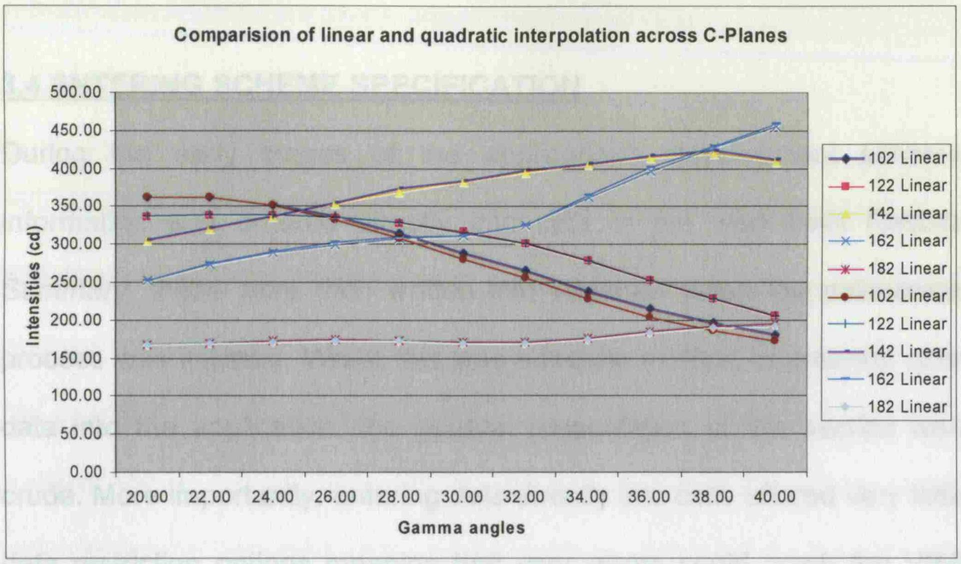



*Figure 3.3.5*

*Chart showing the intensities generated from linear and quadratic interpolation on measured and un-measured angles of a road optic across Gamma angles.*

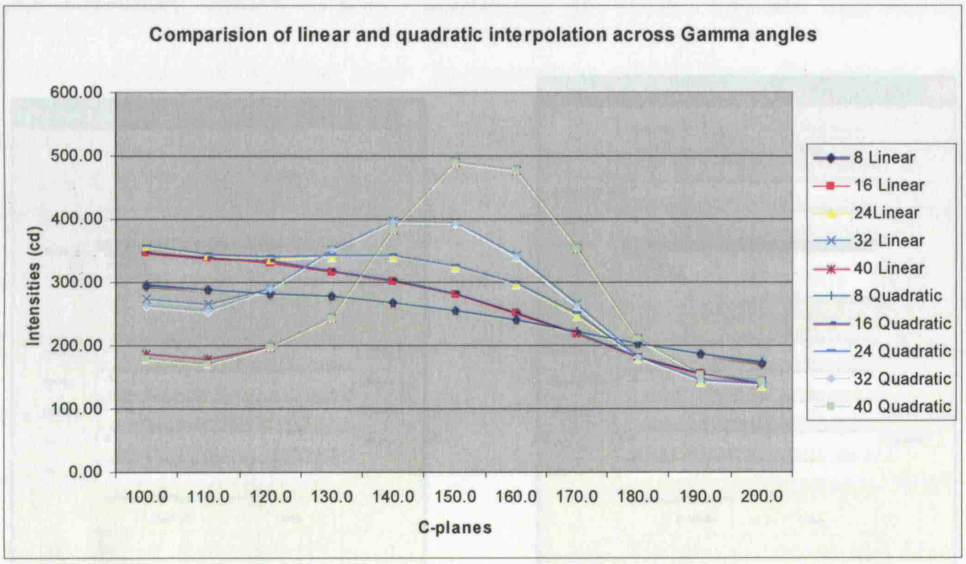



*Figure 3.3.6*

*Chart showing the intensities generated from linear and quadratic interpolation on measured and un-measured angles of a road optic across C-planes.*

## 3.4 ENTERING SCHEME SPECIFICATION

During the early stages of the application's development scheme information was entered directly into cells in the worksheet *Results Summary*, these were then written into variables when the calculation process was initiated. Whilst this was adequite method of entering user data into the application, the general presentation of the section was crude. More importantly, entering data directly into cells offered very little data restriction options meaning that user errors could crash the VBA project. These issues were resolved by creating two separate userforms called *Scheme_Editor_ILL* and *Scheme_Editor_LUM* to enter the scheme specifications for illuminance and luminance calculations as shown in figure 3.4.1.

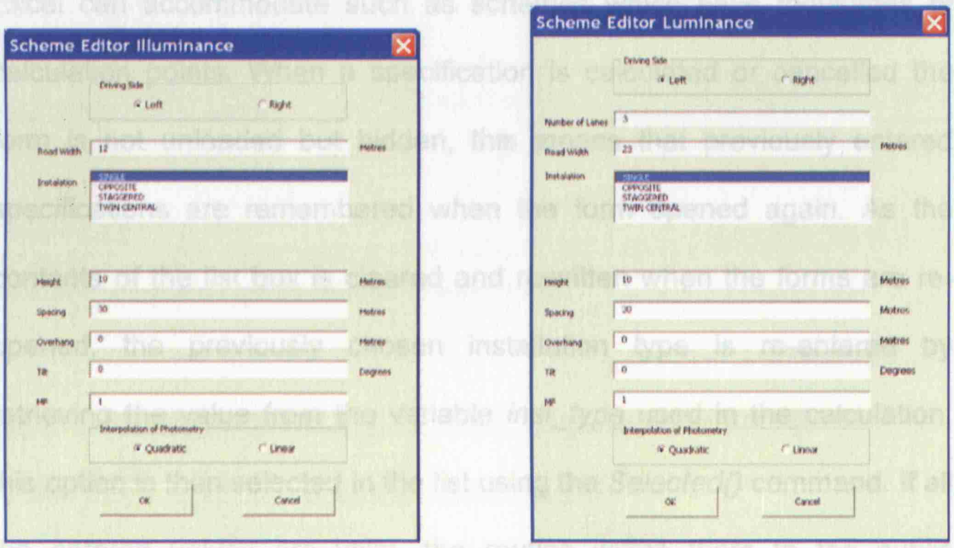



***Figure 3.4.1***

The userforms consist *of textboxes, listboxes* and *radiobuttons* to collect the data; numerical data such as road width and column height are entered in textboxes, the installation type is chosen from a listbox and the method of interpolation and driving side are chosen from radio buttons. Whilst the interface of listboxes and radiobuttons ensure that a valid option is inputted, the values entered into the textboxes are checked to ensure they are within the program's confines. When a user clicks the *OK* button the routine checks that the values are numerical using an *If Is Not Numeric* statement, if this is true code highlights the invalid data and exits the sub. The routine has restrictions on minimum and maximum values to avoid scenarios generated beyond that which Excel can accommodate such as schemes which have thousands of calculation points. When a specification is calculated or cancelled the form is not unloaded but hidden, this means that previously entered specifications are remembered when the form opened again. As the contents of the list box is cleared and rewritten when the forms are re-opened, the previously chosen installation type is re-entered by retrieving the value from the variable *inst_type* used in the calculation, this option is then selected in the list using the *Selected()* command. If all the entered values are valid, the routine writes them to the public variables used to define to calculations components and triggers the *Collect_Workbook_*Photometry routine; the names assigned to the form components are shown in figure 3.4.2. These restrictions and enhancements have been added and developed throughout the applications development as it was unrealistic to anticipate all potential errors at the time of creation. The code for these forms are contained in the *Scheme_Editor* form modules and sample code showing how the entered values are checked for errors is listed in appendix E (page 97).

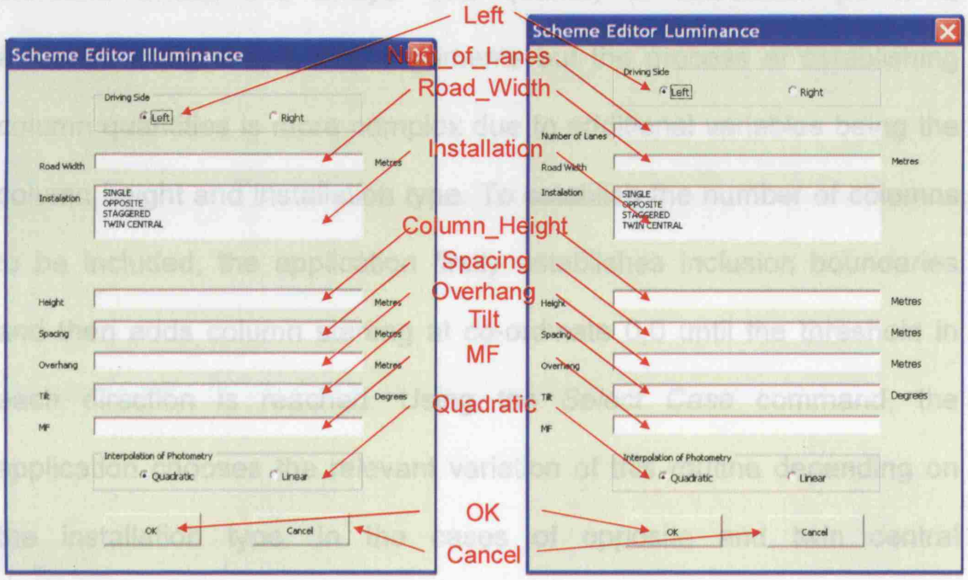



***Figure 3.4.2***

*Allocated component names of scheme editor forms*

## 3.5 ESTABLISHING CALCULATION POINT, COLUMN AND OBSERVER QUANTITIES AND LOCATIONS

The first routine activated in the calculation process is *Collect_Workbook_Photometry*; this is the routine which establishes the quantities of calculation points, columns and, when calculating luminance, observer locations. As the results of this procedure form the basis of all further calculations in the application, these values are shown in the beginning part of the *Results Details* worksheet shown in figure 3.5.1, so that user can check these figures. This routine is short as it simply calls the functions *Num_Of_Columns, num_of_trans_points* and *num_of_long_points* to calculate the number of columns and calculation pointsusing the conventions stated as shown in figure 3.5.2 and 3.5.3and written into arrays. The quantity of calculation points is achieved using simple loop arguments but the process of establishing column quantities is more complex due to additional variables being the column height and installation type. To establish the number of columns to be included, the application firstly establishes inclusion boundaries and then adds column starting at co-ordinate 0,0 until the threshold in each direction is reached. Using the *Select Case* command, the application chooses the relevant variation of this routine depending on the installation type. In the cases of opposite and twin central arrangements a single row can be doubled to return the total value but as the column locations in a staggered arrangement vary on each side of the road each row is computed separately.

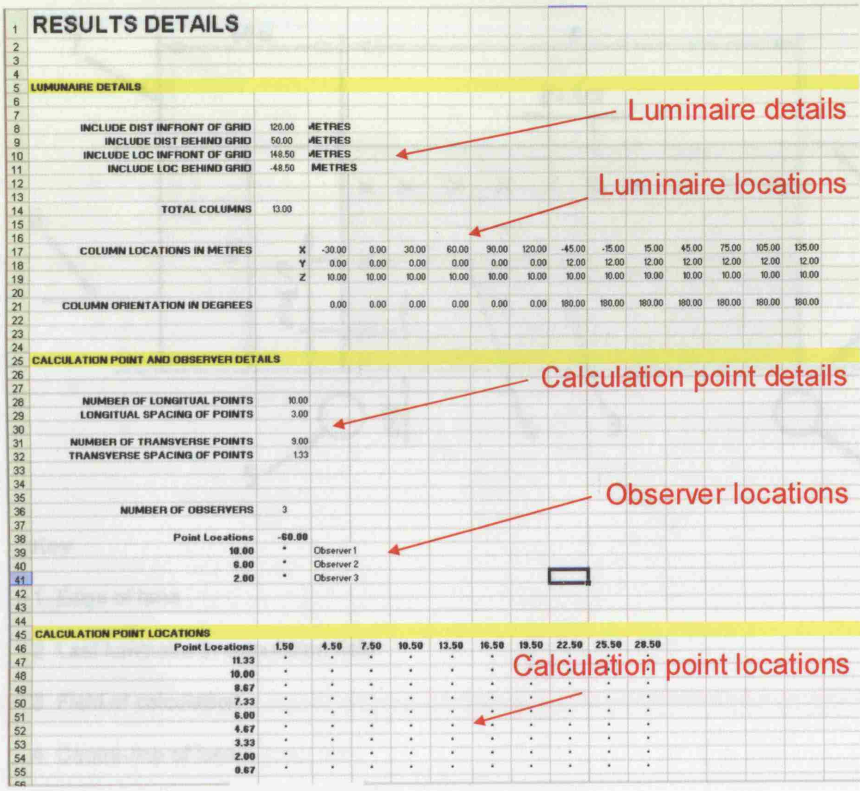
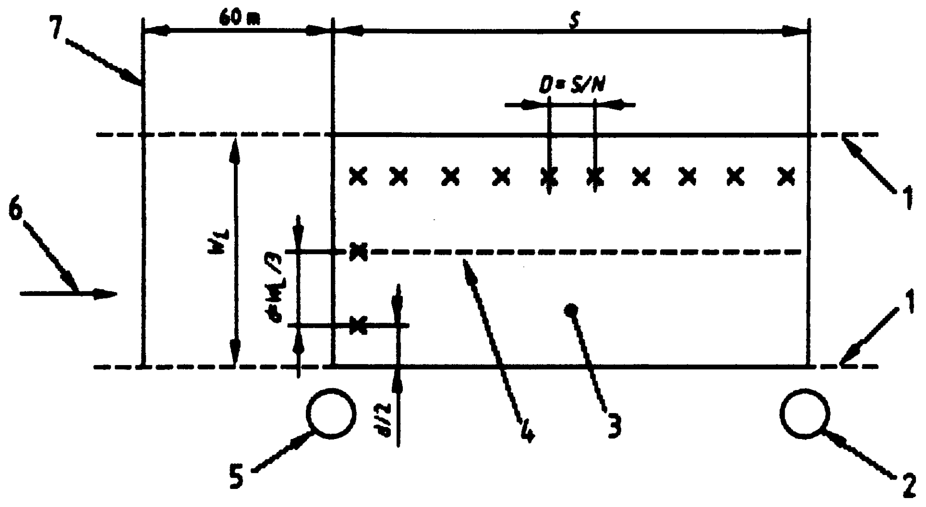



***Figure 3.5.1***

*Top section of the Results Details worksheet displaying the initial scheme data.*




**Key**

1 Edge of lane

2 Last luminaire in calculation field

3 Field of calculation

4 Centre-line of lane

5 First luminaire in calculation field

6 Observation direction

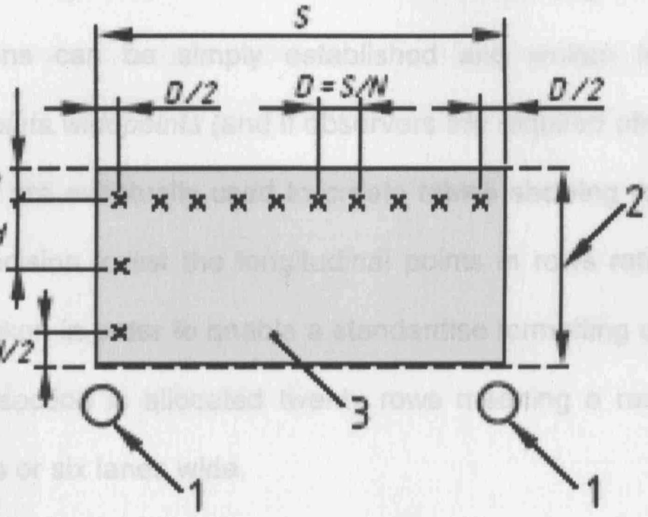
7 Observer's longitudinal position

X denotes lines of calculation points in the transverse and longitudinal directions.

***Figure 3.5.2***

*Convention for the location of calculation points for luminance calculations*[12].

[12] *EN 13201-3.* BSI (2004)




Key

1 Luminaire

2 Width of relevant area $W_r$

3 Field of calculation

x denotes lines of calculation points in the transverse and longitudinal directions

*Figure 3.5.3*

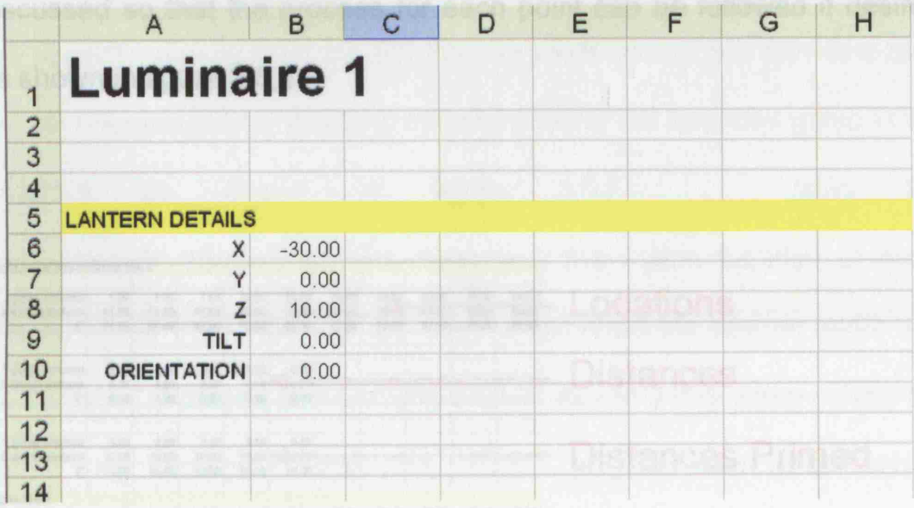
*Convention for the location of calculation points for illuminance calculations*[13].

With the calculation point, column and observer quantities defined, the routine *Write_Calculation_Points* is begun. This is the largest routine in the application was its function extends as far as calculating the lighting values per column. Its first procedure however is to establishes the locations of the scheme components; the co-ordinate system used denotes X, Y, Z as the world axis and allocates absolute zero at the location of the edge of the road in front of the grid. As the number of

[13] *EN 13201-3.* BSI (2004)

calculation points, road width and column spacing are known the point locations can be simply established and written into arrays being *longpoints,widepoints* (and if observers are required *obs_trans_loc)*. This arrays are eventually used to create tables showing various values and the decision to list the longitudinal points in rows rather than columns was taken in order to enable a standardise formatting of the worksheets. Each section is allocated twenty rows meaning a road can be twelve metres or six lanes wide.

As the volume of code to establish the column locations is so large it is contained in the function *Write_Column_Worksheets,* this is called after the calculation point arrays have been written. As the principle of the application is to allocate a worksheet per luminaire, the function's first process is to delete any worksheets that might be present from previous calculations. Using the *worksheets.count* command the application loops through the worksheet nams and deletes all worksheets with names other than *Results Summary, Results Details, Photometry, R-table* and *Luminaire1. Luminaire1* is not deleted was it is used as a template by clearing the contents and duplicating the additional worksheets from it. Once the necessary number of worksheets exists in the workbook a loop procedure calls each worksheet and writes the name, location, orientation and tilt of a column into the first rows as shown in figure 3.5.2. This procedure always starts with column located furthest left and continues to duplicate columns at the set spacing until the threshold value is reached. The code establishing all these values is contained in the *Write_Calculation_Points*, *Col_Copy_Locations* and *Functions* modules and sample code showing how the quantities of columns, calculation points and observers was established is listed in appendix F (page 105)



| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | **Luminaire 1** | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | LANTERN DETAILS | | | | | | | |
| 6 | X | -30.00 | | | | | | |
| 7 | Y | 0.00 | | | | | | |
| 8 | Z | 10.00 | | | | | | |
| 9 | TILT | 0.00 | | | | | | |
| 10 | ORIENTATION | 0.00 | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | |

***Figure 3.5.2***

*Top rows of a sample Luminaire1 work sheet*

## 3.6 ILLUMINANCE

Once the columns are located and aimed the application has all the information necessary to calculate the lighting values. Whilst the process for calculating both luminance and illuminance are similar, illuminance was developed first as this is the simpler calculation of the two. After the routine *Write_Column_Worksheets* is complete VBA returns to the *Write_Calculation_Points* routine and begins to process the data required to generate an illuminance value for each luminaire. The application follows the chronological stages of the calculation which are establishing the distance between the column and point, calculating the angles, interpolating the required intensity and using the above to calculate the illuminance. This process is performed on each luminaire so that the results on each worksheet are presented in isolation. The calculated data is presented using the tabular format previously discussed so that the process for each point can be followed if desired as shown in figure 3.6.2.

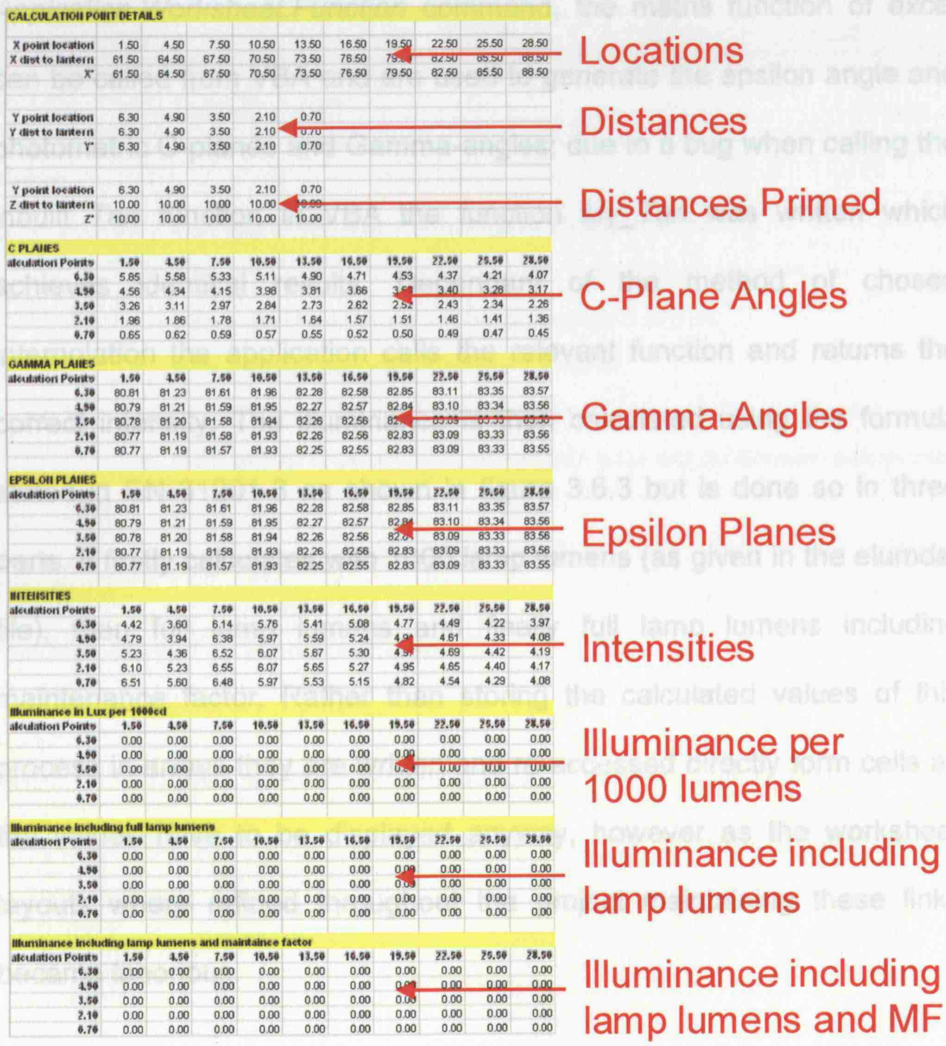


***Figure 3.6.1***

*Presented calculation process for illuminance calculations.*

The process begins by establishing the relationship of the column and calculation point location; this calculated as the longitudinal distance (X), the transverse distance (Y), and the height distance (Z). In order to call the correct intensity from the photometry these values are primed based on the orientation and tilt of the luminaire using the formulas given in EN 13201-3 as shown in figure 3.6.2. Using the *Appication.Worksheet.Function* command, the maths function of excel can be called from VBA and are used to generate the epsilon angle and photometric C-planes and Gamma-angles; due to a bug when calling the inbuilt *Tan* function in VBA the function *My_Tan* was written which achieves identical results. Dependant of the method of chosen interpolation the application calls the relevant function and returns the correct intensity. The illuminance is then calculated using the formula stated in EN 31201-3 as shown in figure 3.6.3 but is done so in three parts. It firstly calculates with 1000 lamp lumens (as given in the elumdat file), then full lamp lumens and finally full lamp lumens including maintenance factor. Rather than storing the calculated values of this process in arrays they are written and re-accessed directly form cells as the values have to be displayed anyway, however as the worksheet layouts where refined throughout the project maintaining these links became laborious.

**6.4 Calculation of *C* and *γ***

These can be determined in four stages:

1) Substitution of $v$, $\psi$, $\delta$, $x$ and $y$ in the equations:

$$x' = x(\cos v \ \cos\psi - \sin v \ \sin\delta \ \sin\psi) + y(\sin v \ \cos\psi - \cos v \ \sin\delta \ \sin\psi) + H \ \cos\delta \ \sin\psi \tag{21}$$

$$y' = -x \ \sin v \ \cos\delta + y \ \cos v \ \cos\delta - H \ \sin\delta \tag{22}$$

$$H' = -x(\sin v \ \sin\delta \ \cos\psi + \cos v \ \sin\psi) - y(\sin v \ \sin\psi - \cos v \ \sin\delta \ \cos\psi) + H \ \cos\delta \cos\psi \tag{23}$$

where:

$x$ and $y$ are the longitudinal and transverse distances between the calculation point and the nadir of the luminaire in Figure 6

$H$ is the height of the luminaire above the calculation point

***Figure 3.6.2***

*Formulas required to calculate primed values for X, Y, Z distances with correction*[14].

$$E = \frac{I \times \cos^3\varepsilon \times \Phi \times MF}{H^2} \tag{35}$$

where:

$E$ is the maintained horizontal illuminance at the point, in lux

$I$ is the intensity in the direction of the point, in candelas per kilolumen

$\varepsilon$ is the angle of incidence of the light at the point, in degrees

$H$ is the mounting height of the luminaire, in metres

$\Phi$ is the initial luminous flux of the lamp or lamps in the luminaire, in kilolumens

$MF$ is the product of the lamp flux maintenance factor and the luminaire maintenance factor

***Figure 3.6.3***

*Formulas required to calculate illuminance*[15].

Major errors such as mis-communication between cells where identified by comparing the results generated of sample columns to that of the

[14] *CIE140* (2000)
[15] *EN 13201-3.* BSI (2004)

established calculation engine calculux[16]; this was obviously not used as a method of ultimate verification but proved an efficient process of debugging obvious errors in the code. The values where entered into tables and plotted onto charts as shown figure 3.6.4, tables and full size charts are shown in appendix H (page115). The code used to calculate illuminance is contained in the Write_*Calc_Points* module and the process is listed in appendix G (page 109).

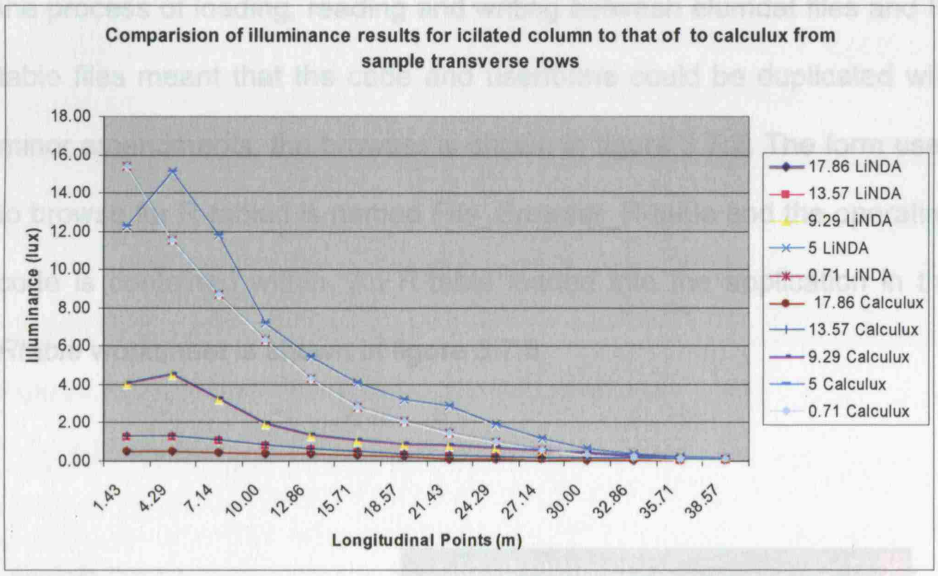



***Figure 3.6.4***

*Comparative chart of illuminance results from application and Calculux.*

## 3.7 R-TABLE

To calculate the achieved luminance of an installation the photometric qualities of the road surface need to be known, these are given in R-tables. The application has been designed to read the R-table files of

16 http://www.lighting.philips.com/gl_en/tools_downloads/calculuxdialux/downloads.php?main=global&parent=4390&id=gl_en_calculuxdialux&lang=en (2006)

Calculux as they are similar in structure to elumdat files and can be easily acquired by downloading the Calculux program. As with elumdat files, R-tables can be opened using notepad and have the same basic layout although there is far less general data as shown in figure 3.7.1. Of particular importance is the *Q0 value* listed at the beginning of the file, as this states the overall reflectivity of the road surface and all values listed in the file need to be multiplied by this figure. The similarities between the process of loading, reading and writing between elumdat files and R-table files meant that the code and userforms could be duplicated with minor amendments; the browser is shown in figure 3.7.2. The form used to browse for R-tabled is named *File_Browser_*R-table and the operating code is contained within. An R-table loaded into the application in the *Rtable* worksheet is shown in figure 3.7.3.

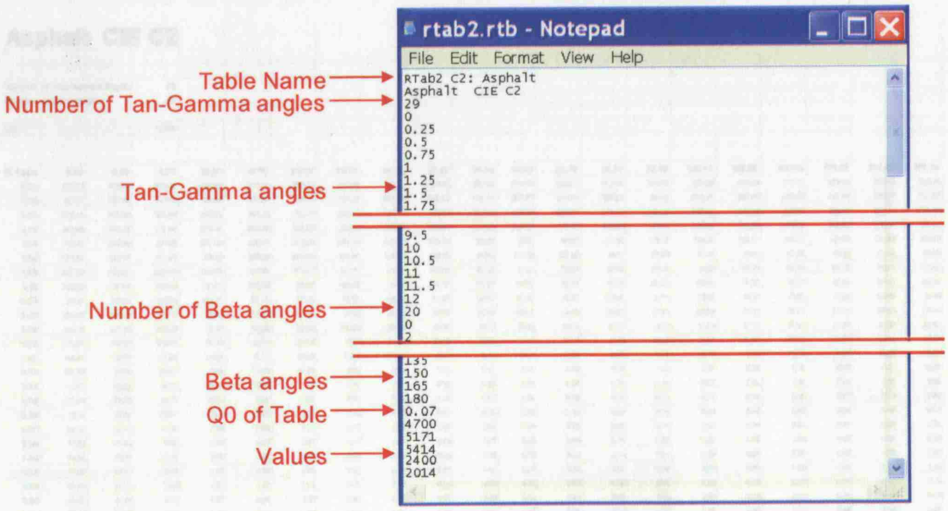



***Figure 3.7.1***

*Sample R-Table opened in Notepad.*

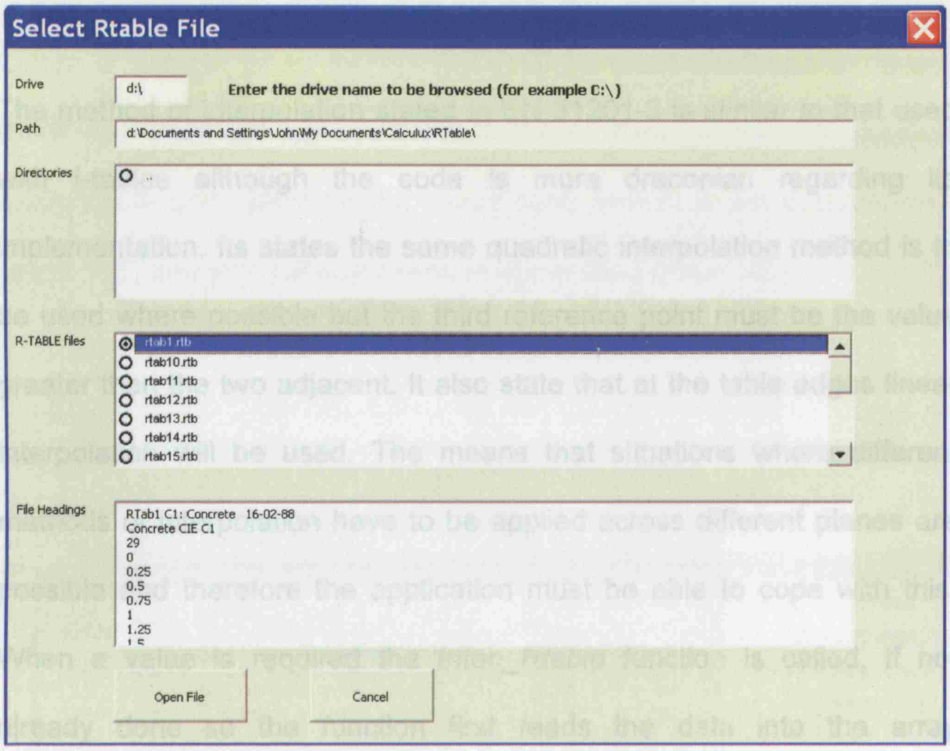



***Figure 3.7.2***

*R-Table file browser.*

**Asphalt CIE C2**

| | |
|---|---|
| Number of Tan Gamma Angles | 29 |
| Number of Beta Angles | 20 |
| Q0 | 0.07 |

| R-Table | 0.00 | 2.00 | 5.00 | 10.00 | 15.00 | 20.00 | 25.00 | 30.00 | 35.00 | 40.00 | 45.00 | 60.00 | 75.00 | 90.00 | 105.00 | 120.00 | 135.00 | 150.00 | 165.00 | 180.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 | 329.00 |
| 0.25 | 361.97 | 357.98 | 371.00 | 364.00 | 371.00 | 368.97 | 361.97 | 357.00 | 350.98 | 349.02 | 347.97 | 339.99 | 328.02 | 311.99 | 298.97 | 294.00 | 297.99 | 287.98 | 291.97 | 280.98 |
| 0.50 | 378.98 | 367.99 | 374.99 | 373.03 | 367.01 | 359.03 | 350.00 | 339.99 | 328.02 | 317.03 | 305.97 | 280.00 | 266.00 | 248.99 | 237.02 | 237.02 | 231.00 | 231.00 | 227.01 | 234.99 |
| 0.75 | 380.03 | 374.99 | 378.00 | 364.98 | 350.98 | 333.97 | 315.00 | 294.98 | 275.03 | 255.99 | 238.98 | 217.98 | 198.03 | 178.01 | 175.00 | 175.98 | 175.98 | 168.98 | 175.00 | 175.98 |
| 1.00 | 371.98 | 374.99 | 371.98 | 353.99 | 315.00 | 276.99 | 242.97 | 220.99 | 205.03 | 192.01 | 181.02 | 151.97 | 133.98 | 129.99 | 125.02 | 123.97 | 125.02 | 129.01 | 128.03 | 128.03 |
| 1.25 | 374.99 | 373.03 | 352.03 | 318.01 | 265.02 | 220.99 | 189.00 | 165.97 | 150.01 | 136.01 | 125.02 | 107.03 | 91.00 | 93.03 | 91.00 | 91.00 | 87.99 | 94.01 | 97.02 | 97.02 |
| 1.50 | 353.99 | 352.03 | 336.00 | 270.97 | 213.01 | 170.03 | 140.00 | 121.03 | 108.99 | 97.02 | 87.01 | 76.02 | 66.99 | 65.03 | 66.01 | 66.01 | 66.99 | 67.97 | 70.98 | 70.98 |
| 1.75 | 332.99 | 326.97 | 301.98 | 221.97 | 165.97 | 129.01 | 104.02 | 90.02 | 74.97 | 67.97 | 63.00 | 52.99 | 51.03 | 49.00 | 49.00 | 46.97 | 52.01 | 51.03 | 52.99 | 53.97 |
| 2.00 | 318.01 | 310.03 | 266.00 | 179.97 | 121.03 | 90.02 | 74.97 | 62.02 | 53.97 | 49.98 | 48.02 | 39.97 | 39.97 | 38.01 | 38.01 | 38.01 | 41.02 | 41.02 | 42.98 | 45.01 |
| 2.50 | 268.03 | 262.01 | 205.03 | 119.00 | 72.03 | 49.98 | 41.02 | 35.98 | 32.97 | 28.98 | 25.97 | 24.99 | 23.03 | 24.01 | 24.99 | 24.01 | 25.97 | 27.02 | 28.98 | 28.00 |
| 3.00 | 227.01 | 217.00 | 147.00 | 73.99 | 42.00 | 28.98 | 24.99 | 23.03 | 21.00 | 18.97 | 17.99 | 16.03 | 16.03 | 17.01 | 17.99 | 17.01 | 18.97 | 21.00 | 21.00 | 23.03 |
| 3.50 | 193.97 | 168.00 | 105.98 | 46.97 | 30.03 | 21.98 | 17.01 | 14.00 | 13.02 | 11.97 | 11.97 | 10.99 | 10.01 | 10.99 | 11.97 | 13.02 | 14.98 | 14.00 | 14.98 | 14.00 |
| 4.00 | 168.00 | 136.01 | 76.02 | 34.02 | 18.97 | 14.00 | 13.02 | 10.99 | 10.01 | 10.01 | 10.01 | 7.98 | 7.98 | 9.03 | 10.01 | 9.03 | 10.99 | 11.97 | 10.99 | 13.02 |
| 4.50 | 140.98 | 111.02 | 53.97 | 21.00 | 14.00 | 10.99 | 9.03 | 7.98 | 7.98 | 7.98 | 7.98 | 7.00 | 7.00 | 7.98 | 7.98 | 7.98 | 7.98 | 10.01 | 10.01 | 10.99 |
| 5.00 | 126.00 | 90.02 | 42.98 | 17.01 | 10.01 | 7.98 | 7.98 | 7.00 | 6.02 | 6.02 | 7.00 | 6.02 | 7.00 | 6.02 | 6.02 | 7.00 | 7.98 | 7.98 | 7.98 | 9.03 |
| 5.50 | 107.03 | 79.03 | 31.99 | 11.97 | 7.98 | 7.00 | 7.00 | 7.00 | 6.02 | 4.97 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6.00 | 94.01 | 65.03 | 25.97 | 10.01 | 7.00 | 6.02 | 6.02 | 6.02 | 4.97 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6.50 | 86.03 | 56.00 | 21.00 | 7.98 | 7.00 | 6.02 | 4.97 | 4.97 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7.00 | 77.98 | 49.98 | 17.01 | 7.00 | 4.97 | 4.97 | 4.97 | 4.97 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7.50 | 70.00 | 41.02 | 14.00 | 7.00 | 3.99 | 3.01 | 3.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8.00 | 63.00 | 37.03 | 10.99 | 4.97 | 3.99 | 3.99 | 3.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8.50 | 59.99 | 37.03 | 10.01 | 4.97 | 3.99 | 3.99 | 3.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 9.00 | 56.00 | 31.99 | 9.03 | 4.97 | 3.99 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 9.50 | 52.99 | 28.00 | 9.03 | 3.99 | 3.99 | 3.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 52.01 | 27.02 | 7.00 | 4.97 | 3.99 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.50 | 45.01 | 23.03 | 7.00 | 3.99 | 3.01 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 11.00 | 42.98 | 21.98 | 7.00 | 3.01 | 3.01 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 11.50 | 52.99 | 21.98 | 7.00 | 3.01 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 12.00 | 42.00 | 20.02 | 7.00 | 3.99 | 3.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

***Figure 3.7.3***

*Sample R-table loaded into the application*

The method of interpolation stated in EN 31201-3 is similar to that used with I-tables although the code is more draconian regarding its implementation. Its states the same quadratic interpolation method is to be used where possible but the third reference point must be the value greater than the two adjacent. It also state that at the table edges linear interpolation will be used. The means that situations where different methods of interpolation have to be applied across different planes are possible and therefore the application must be able to cope with this. When a value is required the *Inten_Rtable* function is called, if not already done so the function first reads the data into the array *Intensity_Rtable* and then finds the three required reference points for interpolation. A *Select Case* structure directs the code to the necessary routine based on the area of the R-table being used, this then calls the *Quadratic_I3* and *Linear_Interpolation* function to generate a value. As with photometric files, ranges for every possible angle are not stated; Beta plane angles are given up to hundred and eighty degrees and Tan-Gamma angles are given up to twelve degrees. No additional formulas are required to cope with this as the method of calculation for the beta plane is done so in a manner that the required angle is always within this range and if the Tan-Gamma angle is greater than twelve the value is considered as zero. In order to verify of the accuracy of the *Inten_R-table* function the same method was used as with the I-Table as shown in figure 3.7.4 and 3.7.5 but with care taken to position sample planes on minimum, middle and maximum planes in order to test all cases. Unmeasured values where then plotted with measured values but as before in sample areas testing all cases as shown in figure 3.7.6 and 3.7.7. The code for this section is in the *R_Table_Interpolator* module and sample code showing the select case section is list in appendix I (page 116) and with full size charts in appendix J (page 122).

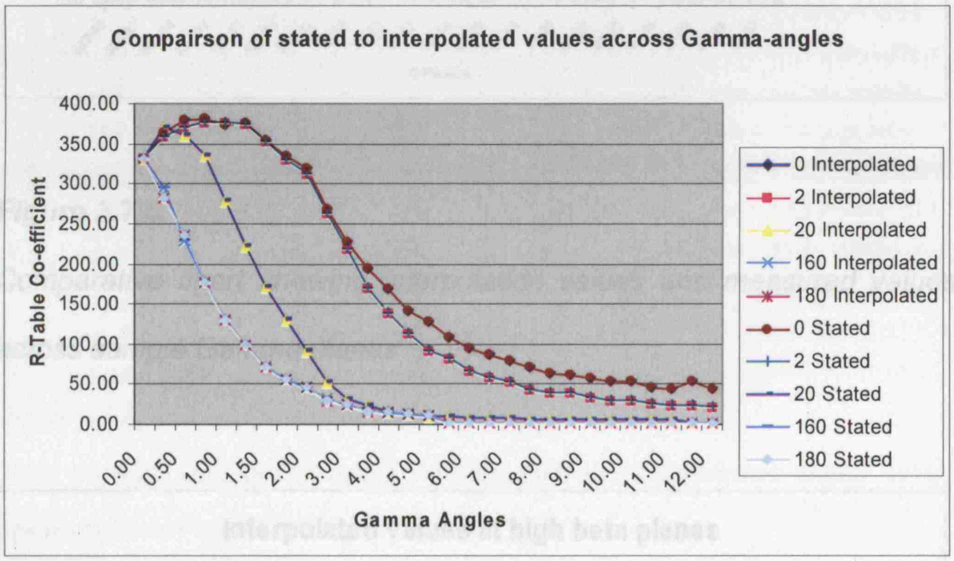



***Figure 3.7.4***

*Comparative chart showing interpolation values and measured values across sample Beta planes*

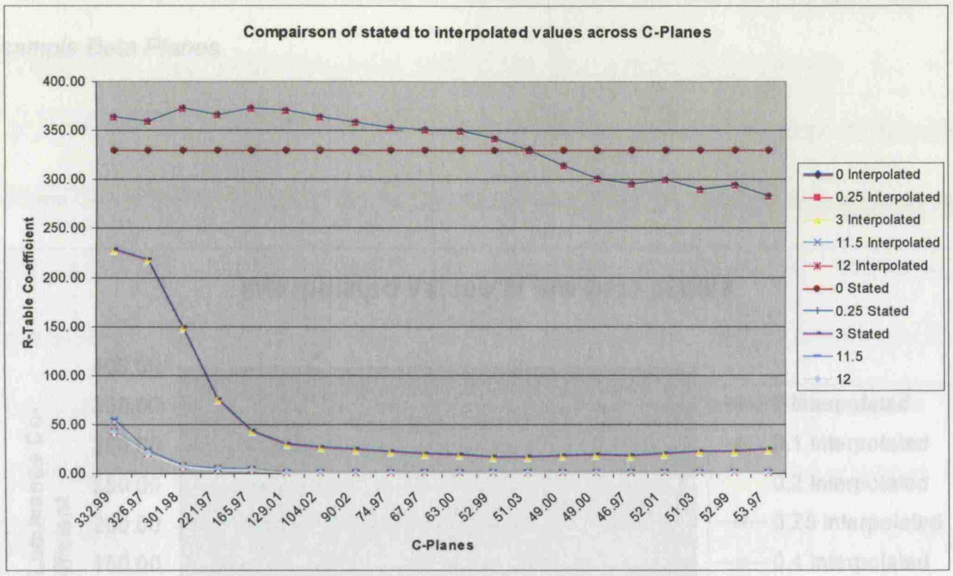



*Figure 3.7.5*

*Comparative chart showing interpolation values and measured values across sample Gamma planes*

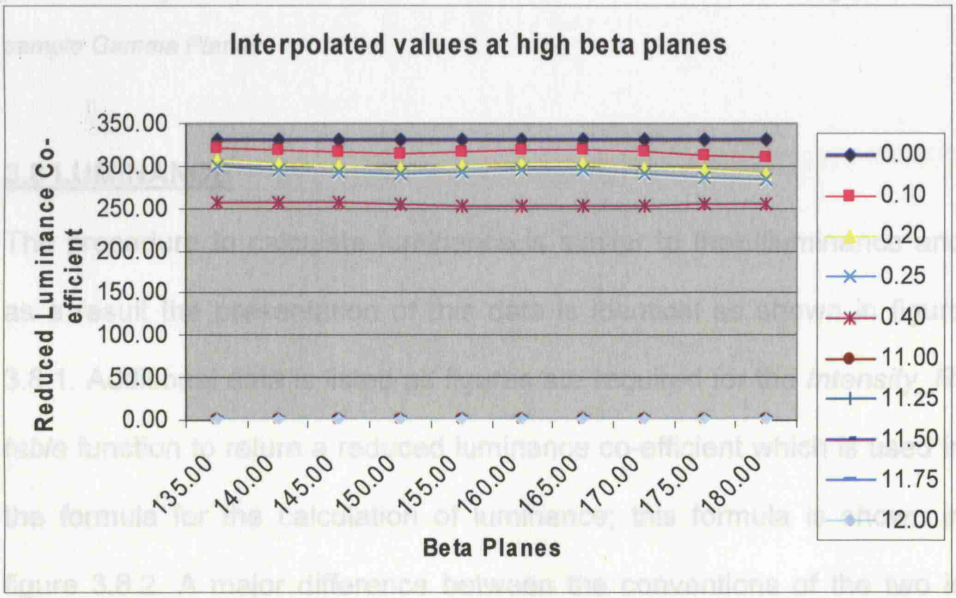



*Figure 3.7.6*

*Table showing interpolated values of measured and un-measured angles across sample Beta Planes*

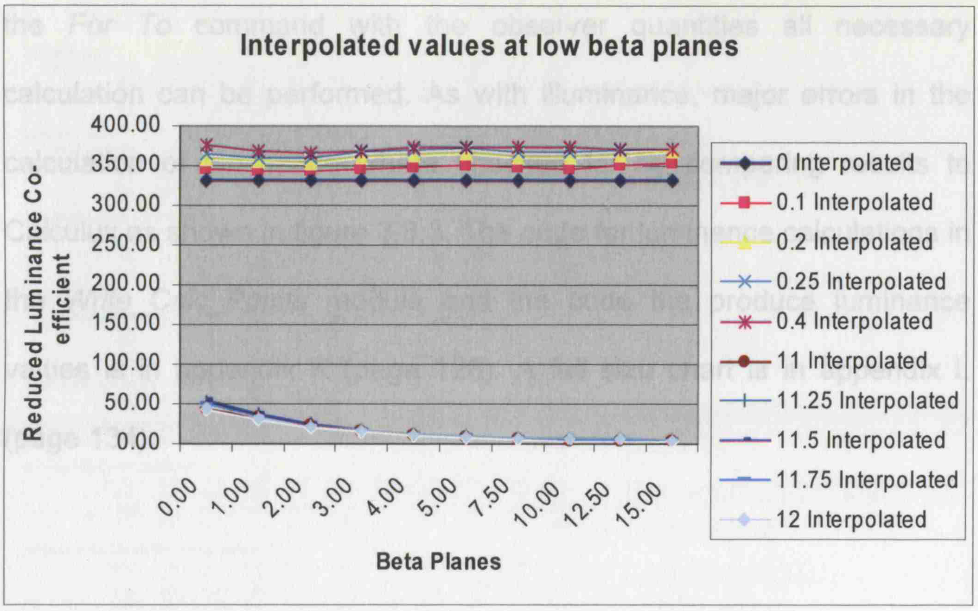
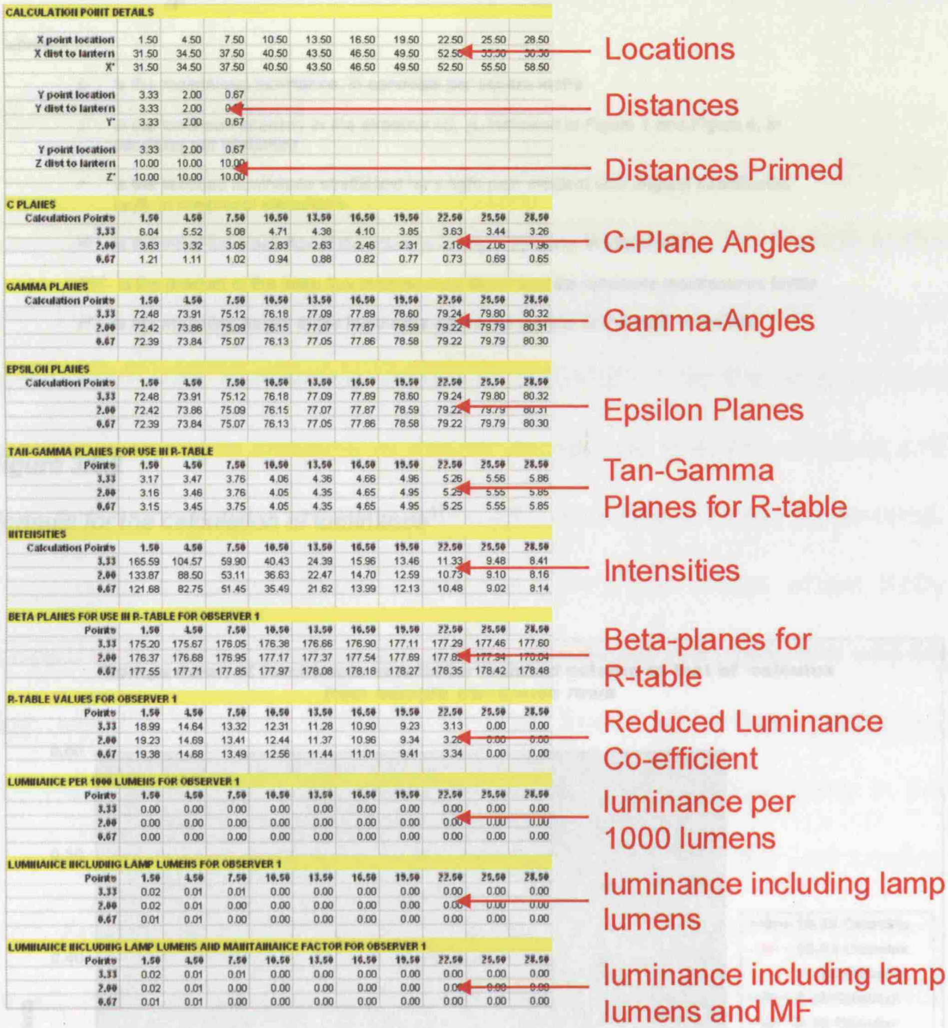



***Figure 3.7.7***

*Table showing interpolated values of measured and un-measured angles across sample Gamma Planes*

## 3.8 LUMINANCE

The procedure to calculate luminance is similar to that illuminance and as a result the presentation of this data is identical as shown in figure 3.8.1. Additional data is listed as figures are required for the *Intensity_R-table* function to return a reduced luminance co-efficient which is used in the formula for the calculation of luminance; this formula is shown in figure 3.8.2. A major difference between the conventions of the two is that luminance calculations require values at each point to be generated for all observers. This meant that a degree of flexibility within the worksheet structure was required so that calculations for any number of observers are possible and the data be presented clearly. As the worksheet data is laid out in sections which always consist of twenty rows the total number of rows per observer can be calculated and using the *For To* command with the observer quantities all necessary calculation can be performed. As with illuminance, major errors in the calculation of luminance where checked for by comparing results to Calculux as shown in figure 3.8.3. The code for luminance calculations in the *Write_Calc_Points* module and the code the produce luminance values is in appendix K (page 126). A full size chart is in appendix L (page 134).




| CALCULATION POINT DETAILS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| X point location | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
| X dist to lantern | 31.50 | 34.50 | 37.50 | 40.50 | 43.50 | 46.50 | 49.50 | 52.50 | 55.50 | 58.50 |
| X' | 31.50 | 34.50 | 37.50 | 40.50 | 43.50 | 46.50 | 49.50 | 52.50 | 55.50 | 58.50 |
| Y point location | 3.33 | 2.00 | 0.67 | | | | | | | |
| Y dist to lantern | 3.33 | 2.00 | [illegible] | | | | | | | |
| Y' | 3.33 | 2.00 | 0.67 | | | | | | | |
| Y point location | 3.33 | 2.00 | 0.67 | | | | | | | |
| Z dist to lantern | 10.00 | 10.00 | 10.00 | | | | | | | |
| Z' | 10.00 | 10.00 | 10.00 | | | | | | | |
| **C PLANES** | | | | | | | | | | |
| Calculation Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 6.04 | 5.52 | 5.08 | 4.71 | 4.38 | 4.10 | 3.85 | 3.63 | 3.44 | 3.26 |
| **2.00** | 3.63 | 3.32 | 3.05 | 2.83 | 2.63 | 2.46 | 2.31 | 2.16 | 2.06 | 1.96 |
| **0.67** | 1.21 | 1.11 | 1.02 | 0.94 | 0.88 | 0.82 | 0.77 | 0.73 | 0.69 | 0.65 |
| **GAMMA PLANES** | | | | | | | | | | |
| Calculation Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 72.48 | 73.91 | 75.12 | 76.18 | 77.09 | 77.89 | 78.60 | 79.24 | 79.80 | 80.32 |
| **2.00** | 72.42 | 73.86 | 75.09 | 76.15 | 77.07 | 77.87 | 78.59 | 79.22 | 79.79 | 80.31 |
| **0.67** | 72.39 | 73.84 | 75.07 | 76.13 | 77.05 | 77.86 | 78.58 | 79.22 | 79.79 | 80.30 |
| **EPSILON PLANES** | | | | | | | | | | |
| Calculation Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 72.48 | 73.91 | 75.12 | 76.18 | 77.09 | 77.89 | 78.60 | 79.24 | 79.80 | 80.32 |
| **2.00** | 72.42 | 73.86 | 75.09 | 76.15 | 77.07 | 77.87 | 78.59 | 79.22 | 79.79 | 80.31 |
| **0.67** | 72.39 | 73.84 | 75.07 | 76.13 | 77.05 | 77.86 | 78.58 | 79.22 | 79.79 | 80.30 |
| **TAN-GAMMA PLANES FOR USE IN R-TABLE** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 3.17 | 3.47 | 3.76 | 4.06 | 4.36 | 4.66 | 4.96 | 5.26 | 5.56 | 5.86 |
| **2.00** | 3.16 | 3.46 | 3.76 | 4.05 | 4.35 | 4.65 | 4.95 | 5.25 | 5.55 | 5.85 |
| **0.67** | 3.15 | 3.45 | 3.75 | 4.05 | 4.35 | 4.65 | 4.95 | 5.25 | 5.55 | 5.85 |
| **INTENSITIES** | | | | | | | | | | |
| Calculation Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 165.59 | 104.57 | 59.90 | 40.43 | 24.39 | 15.96 | 13.46 | 11.33 | 9.48 | 8.41 |
| **2.00** | 133.87 | 88.50 | 53.11 | 36.63 | 22.47 | 14.70 | 12.59 | 10.73 | 9.10 | 8.16 |
| **0.67** | 121.68 | 82.75 | 51.45 | 35.49 | 21.62 | 13.99 | 12.13 | 10.48 | 9.02 | 8.14 |
| **BETA PLANES FOR USE IN R-TABLE FOR OBSERVER 1** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 175.20 | 175.67 | 176.05 | 176.38 | 176.66 | 176.90 | 177.11 | 177.29 | 177.46 | 177.60 |
| **2.00** | 176.37 | 176.68 | 176.95 | 177.17 | 177.37 | 177.54 | 177.69 | 177.82 | 177.94 | 178.04 |
| **0.67** | 177.55 | 177.71 | 177.85 | 177.97 | 178.08 | 178.18 | 178.27 | 178.35 | 178.42 | 178.48 |
| **R-TABLE VALUES FOR OBSERVER 1** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 18.99 | 14.64 | 13.32 | 12.31 | 11.28 | 10.90 | 9.23 | 3.13 | 0.00 | 0.00 |
| **2.00** | 19.23 | 14.69 | 13.41 | 12.44 | 11.37 | 10.96 | 9.34 | 3.28 | 0.00 | 0.00 |
| **0.67** | 19.38 | 14.68 | 13.49 | 12.56 | 11.44 | 11.01 | 9.41 | 3.34 | 0.00 | 0.00 |
| **LUMINANCE PER 1000 LUMENS FOR OBSERVER 1** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **2.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **0.67** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **LUMINANCE INCLUDING LAMP LUMENS FOR OBSERVER 1** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **2.00** | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **0.67** | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **LUMINANCE INCLUDING LAMP LUMENS AND MAINTAINANCE FACTOR FOR OBSERVER 1** | | | | | | | | | | |
| Points | **1.50** | **4.50** | **7.50** | **10.50** | **13.50** | **16.50** | **19.50** | **22.50** | **25.50** | **28.50** |
| **3.33** | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **2.00** | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **0.67** | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

***Figure 3.8.1***

*Presented calculation process for luminance calculations for a sample observer.*

$$L = \frac{I \times r \times \Phi \times MF \times 10^{-4}}{H^2} \quad (31)$$

where:

- $L$ is the maintained luminance, in candelas per square metre
- $I$ is the luminous intensity in the direction ($C$, $\gamma$), indicated in Figure 1 and Figure 4, in candelas per kilolumen
- $r$ is the reduced luminance coefficient for a light path incident with angular coordinates ($\varepsilon$,$\beta$), in reciprocal steradians
- $\Phi$ is the initial luminous flux of the sources in each luminaire, in kilolumens
- $MF$ is the product of the lamp flux maintenance factor and the luminaire maintenance factor
- $H$ is the mounting height of the luminaires above the surface of the road, in metres

***Figure 3.8.2***

*Formula for the calculation of luminance*[17]

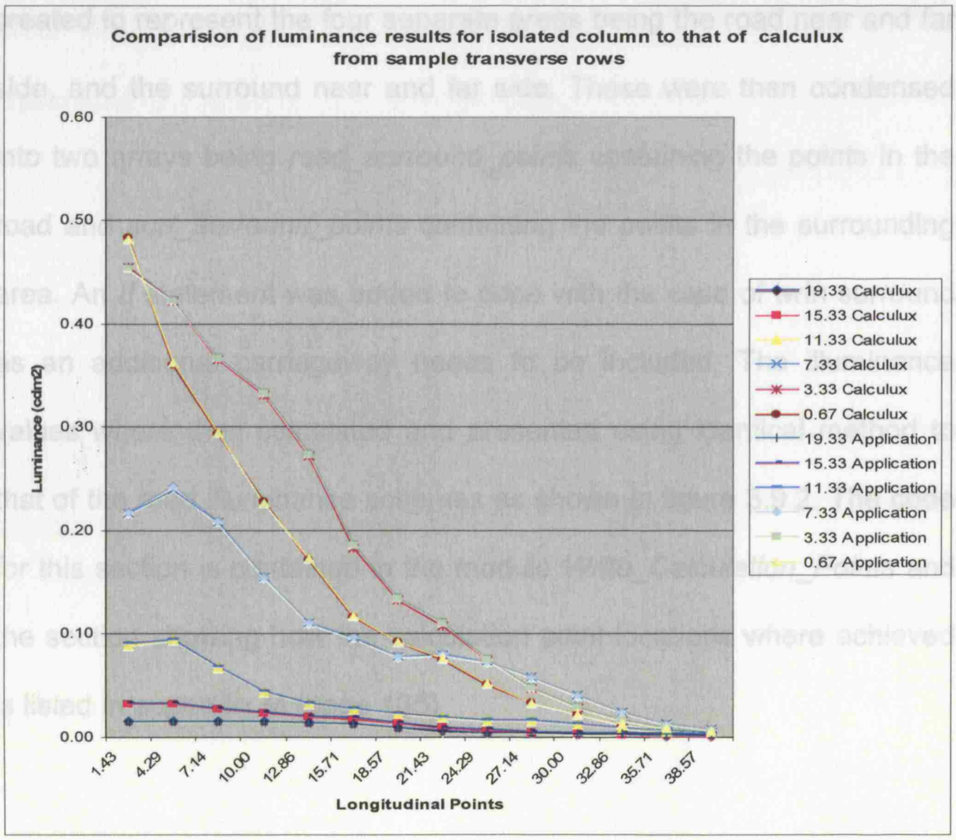


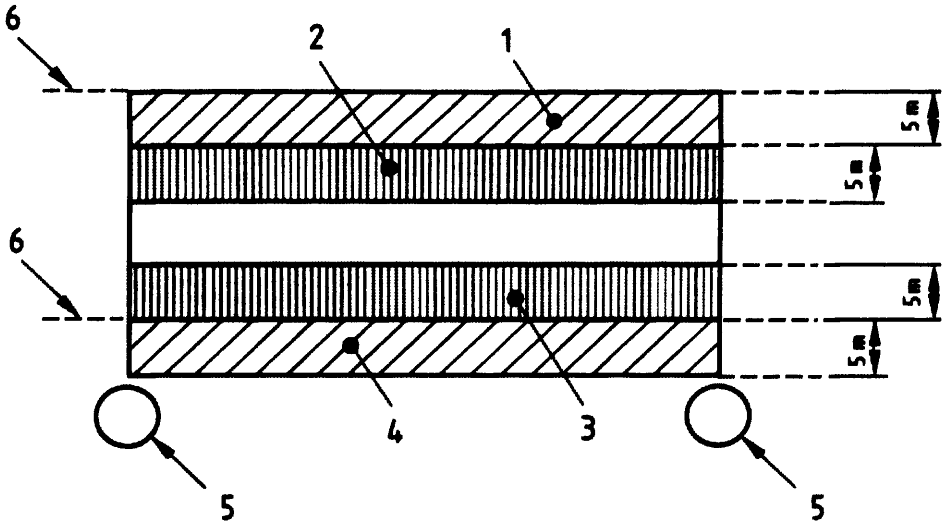

***FIGURE 3.8.3***

[17] *EN 13201-3.* BSI (2004)

*Comparative chart of illuminance results from application and Calculux.*

## 3.9 SURROUND RATIO

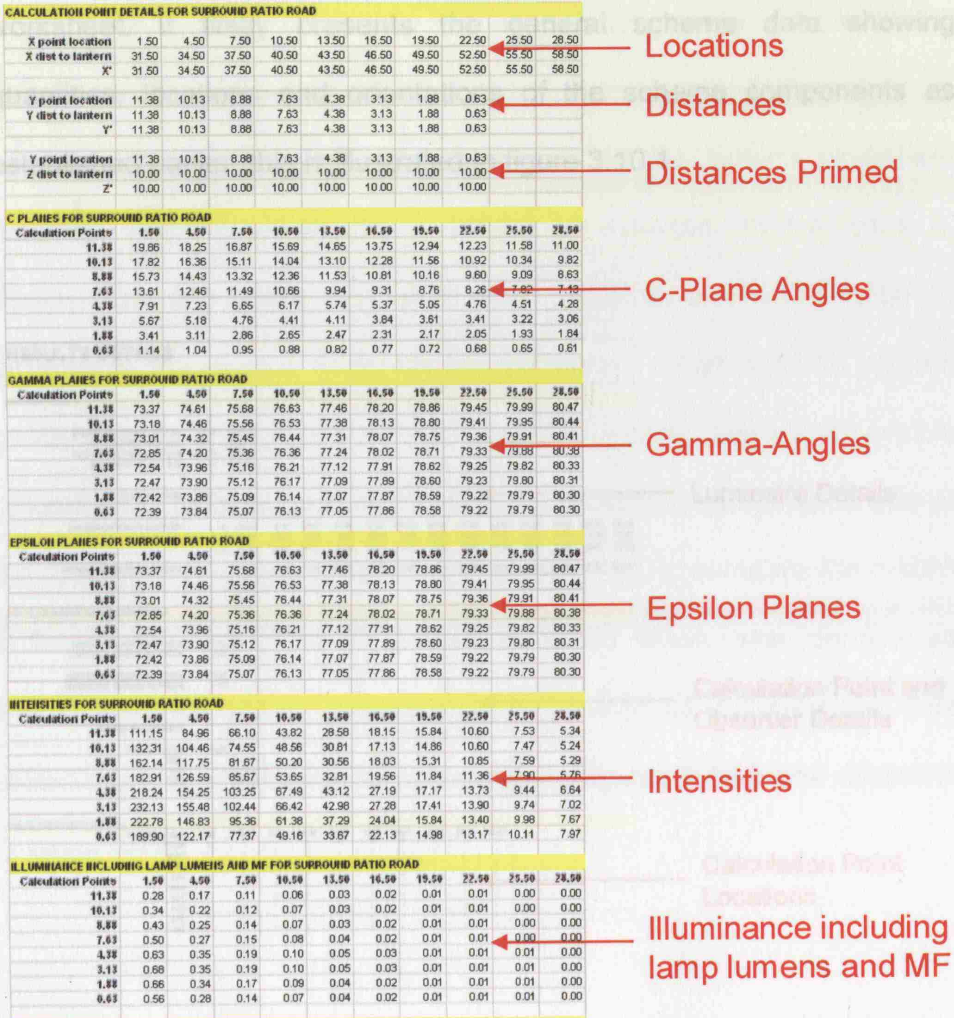
Schemes measured in luminance also require a figure representing the surround ratio to be given; this is a comparison of the illuminance at the road edge to the illuminance of the local surrounding area as shown in figure 3.9.1. The convention stated in EN31021-3 for the spacing and quantity of points is identical to that of illuminance road calculations and as a result the array containing the longitudinal points could be re-used. To generate locations for the transverse points four arrays where firstly created to represent the four separate areas being the road near and far side, and the surround near and far side. These were then condensed into two arrays being *road_surround_points* containing the points in the road and *surr_surround_points* containing the points in the surrounding area. An *If* statement was added to cope with the case of twin surround as an additional carriageway needs to be included. The illuminance values where then calculated and presented using identical method to that of the road illuminance schemes as shown in figure 3.9.2. The code for this section is contained in the module *Write_Calculation_Points* and the section showing how the calculation point locations where achieved is listed in appendix M (page 135).




a) Width of strip equals 5 m

***Figure 3.9.1***

*Conventions for calculation of surround ratio*



CALCULATION POINT DETAILS FOR SURROUND RATIO ROAD

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| X point location | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
| X dist to lantern | 31.50 | 34.50 | 37.50 | 40.50 | 43.50 | 46.50 | 49.50 | 52.50 | 55.50 | 58.50 |
| X' | 31.50 | 34.50 | 37.50 | 40.50 | 43.50 | 46.50 | 49.50 | 52.50 | 55.50 | 58.50 |
| Y point location | 11.38 | 10.13 | 8.88 | 7.63 | 4.38 | 3.13 | 1.88 | 0.63 | | |
| Y dist to lantern | 11.38 | 10.13 | 8.88 | 7.63 | 4.38 | 3.13 | 1.88 | 0.63 | | |
| Y' | 11.38 | 10.13 | 8.88 | 7.63 | 4.38 | 3.13 | 1.88 | 0.63 | | |
| Y point location | 11.38 | 10.13 | 8.88 | 7.63 | 4.38 | 3.13 | 1.88 | 0.63 | | |
| Z dist to lantern | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | | |
| Z' | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | | |

C PLANES FOR SURROUND RATIO ROAD

| Calculation Points | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11.38 | 19.86 | 18.25 | 16.87 | 15.69 | 14.65 | 13.75 | 12.94 | 12.23 | 11.58 | 11.00 |
| 10.13 | 17.82 | 16.36 | 15.11 | 14.04 | 13.10 | 12.28 | 11.56 | 10.92 | 10.34 | 9.82 |
| 8.88 | 15.73 | 14.43 | 13.32 | 12.36 | 11.53 | 10.81 | 10.16 | 9.60 | 9.09 | 8.63 |
| 7.63 | 13.61 | 12.46 | 11.49 | 10.66 | 9.94 | 9.31 | 8.76 | 8.26 | 7.82 | [illegible] |
| 4.38 | 7.91 | 7.23 | 6.65 | 6.17 | 5.74 | 5.37 | 5.05 | 4.76 | 4.51 | 4.28 |
| 3.13 | 5.67 | 5.18 | 4.76 | 4.41 | 4.11 | 3.84 | 3.61 | 3.41 | 3.22 | 3.06 |
| 1.88 | 3.41 | 3.11 | 2.86 | 2.65 | 2.47 | 2.31 | 2.17 | 2.05 | 1.93 | 1.84 |
| 0.63 | 1.14 | 1.04 | 0.95 | 0.88 | 0.82 | 0.77 | 0.72 | 0.68 | 0.65 | 0.61 |

GAMMA PLANES FOR SURROUND RATIO ROAD

| Calculation Points | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11.38 | 73.37 | 74.61 | 75.68 | 76.63 | 77.46 | 78.20 | 78.86 | 79.45 | 79.99 | 80.47 |
| 10.13 | 73.18 | 74.46 | 75.56 | 76.53 | 77.38 | 78.13 | 78.80 | 79.41 | 79.95 | 80.44 |
| 8.88 | 73.01 | 74.32 | 75.45 | 76.44 | 77.31 | 78.07 | 78.75 | 79.36 | 79.91 | 80.41 |
| 7.63 | 72.85 | 74.20 | 75.36 | 76.36 | 77.24 | 78.02 | 78.71 | 79.33 | 79.88 | 80.38 |
| 4.38 | 72.54 | 73.96 | 75.16 | 76.21 | 77.12 | 77.91 | 78.62 | 79.25 | 79.82 | 80.33 |
| 3.13 | 72.47 | 73.90 | 75.12 | 76.17 | 77.09 | 77.89 | 78.60 | 79.23 | 79.80 | 80.31 |
| 1.88 | 72.42 | 73.86 | 75.09 | 76.14 | 77.07 | 77.87 | 78.59 | 79.22 | 79.79 | 80.30 |
| 0.63 | 72.39 | 73.84 | 75.07 | 76.13 | 77.05 | 77.86 | 78.58 | 79.22 | 79.79 | 80.30 |

EPSILON PLANES FOR SURROUND RATIO ROAD

| Calculation Points | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11.38 | 73.37 | 74.61 | 75.68 | 76.63 | 77.46 | 78.20 | 78.86 | 79.45 | 79.99 | 80.47 |
| 10.13 | 73.18 | 74.46 | 75.56 | 76.53 | 77.38 | 78.13 | 78.80 | 79.41 | 79.95 | 80.44 |
| 8.88 | 73.01 | 74.32 | 75.45 | 76.44 | 77.31 | 78.07 | 78.75 | 79.36 | 79.91 | 80.41 |
| 7.63 | 72.85 | 74.20 | 75.36 | 76.36 | 77.24 | 78.02 | 78.71 | 79.33 | 79.88 | 80.38 |
| 4.38 | 72.54 | 73.96 | 75.16 | 76.21 | 77.12 | 77.91 | 78.62 | 79.25 | 79.82 | 80.33 |
| 3.13 | 72.47 | 73.90 | 75.12 | 76.17 | 77.09 | 77.89 | 78.60 | 79.23 | 79.80 | 80.31 |
| 1.88 | 72.42 | 73.86 | 75.09 | 76.14 | 77.07 | 77.87 | 78.59 | 79.22 | 79.79 | 80.30 |
| 0.63 | 72.39 | 73.84 | 75.07 | 76.13 | 77.05 | 77.86 | 78.58 | 79.22 | 79.79 | 80.30 |

INTENSITIES FOR SURROUND RATIO ROAD

| Calculation Points | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11.38 | 111.15 | 84.96 | 66.10 | 43.82 | 28.58 | 18.15 | 15.84 | 10.60 | 7.53 | 5.34 |
| 10.13 | 132.31 | 104.46 | 74.55 | 48.56 | 30.81 | 17.13 | 14.86 | 10.60 | 7.47 | 5.24 |
| 8.88 | 162.14 | 117.75 | 81.67 | 50.20 | 30.56 | 18.03 | 15.31 | 10.85 | 7.59 | 5.29 |
| 7.63 | 182.91 | 126.59 | 85.67 | 53.65 | 32.81 | 19.56 | 11.84 | 11.36 | 7.90 | 5.76 |
| 4.38 | 218.24 | 154.25 | 103.25 | 67.49 | 43.12 | 27.19 | 17.17 | 13.73 | 9.44 | 6.64 |
| 3.13 | 232.13 | 155.48 | 102.44 | 66.42 | 42.98 | 27.28 | 17.41 | 13.90 | 9.74 | 7.02 |
| 1.88 | 222.78 | 146.83 | 95.36 | 61.38 | 37.29 | 24.04 | 15.84 | 13.40 | 9.98 | 7.67 |
| 0.63 | 189.90 | 122.17 | 77.37 | 49.16 | 33.67 | 22.13 | 14.98 | 13.17 | 10.11 | 7.97 |

ILLUMINANCE INCLUDING LAMP LUMENS AND MF FOR SURROUND RATIO ROAD

| Calculation Points | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11.38 | 0.28 | 0.17 | 0.11 | 0.06 | 0.03 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 |
| 10.13 | 0.34 | 0.22 | 0.12 | 0.07 | 0.03 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 |
| 8.88 | 0.43 | 0.25 | 0.14 | 0.07 | 0.03 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 |
| 7.63 | 0.50 | 0.27 | 0.15 | 0.08 | 0.04 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 |
| 4.38 | 0.63 | 0.35 | 0.19 | 0.10 | 0.05 | 0.03 | 0.01 | 0.01 | 0.01 | 0.00 |
| 3.13 | 0.68 | 0.35 | 0.19 | 0.10 | 0.05 | 0.03 | 0.01 | 0.01 | 0.01 | 0.00 |
| 1.88 | 0.66 | 0.34 | 0.17 | 0.09 | 0.04 | 0.02 | 0.01 | 0.01 | 0.01 | 0.00 |
| 0.63 | 0.56 | 0.28 | 0.14 | 0.07 | 0.04 | 0.02 | 0.01 | 0.01 | 0.01 | 0.00 |



***Figure 3.9.2***

*Calculation process for surround ratio*

## 3.10 RESULTS DETAILS

The *Result_Details* worksheet shows the acclimated data of all the *Luminaire* worksheets for so that final values can be inspected and summaries be generated. When the data for the calculations for luminance or illuminance have been calculated the routine

*Write_Results_Details* begins to compile this data in the *Results_Details* worksheet. It firstly presents the general scheme data showing quantities, locations and orientations of the scheme components as established earlier; this is illustrated in figure 3.10.1.

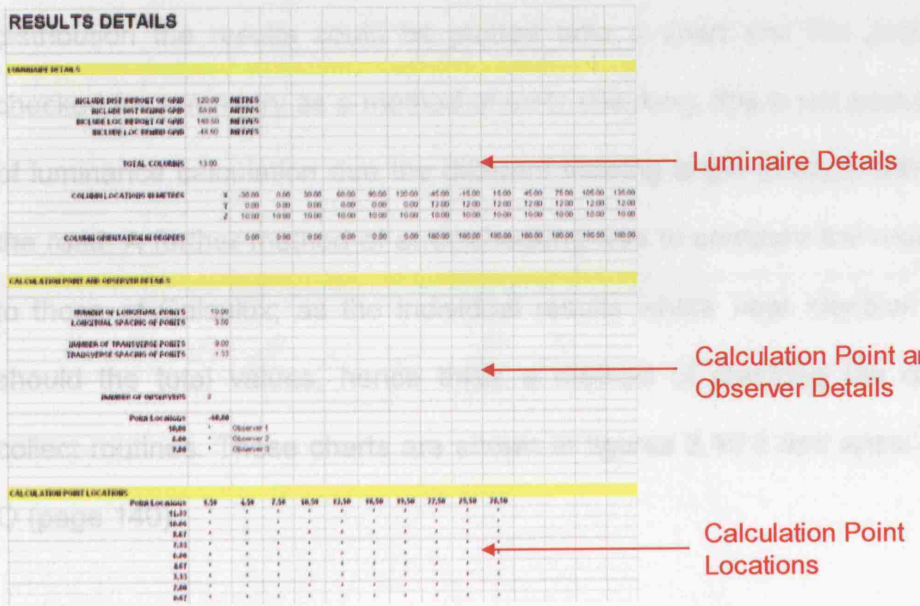



***FIGURE 3.10.1***

*Section of Results Details showing scheme data*

After this data is added a table containing the calculated values including all luminaires in the scheme is written to show total values for each calculation point. This was achieved by creating a structure similar to that of the values form individual luminaires but rather than calculating a value, a loop command was inserted to collect and tally this data. This functions by totalling the values into the variable *illuminance* from each *luminaire* worksheet for each point and then writing them into a table. In the case of luminance calculations where multiple observers needed to be accommodated, the same nested loop structure was used to repeat the process the necessary number of times; this method of referencing cells in other worksheets relative to cells in the current worksheet required a matrix of variables difficult to maintain. In the case of illuminance calculations as most road lanterns are symmetrical in distribution the results could be plotted onto a chart and the pattern checked for symmetry as a method of error checking, this is not possible of luminance calculation due the different viewing angle at each side of the road. A further method of error checking was to compare the results to those of Calculux; as the individual results where near identical so should the total values, hence thisis a method of checking the data collect routines. These charts are shown in figures 3.10.2 and appendix O (page 140).

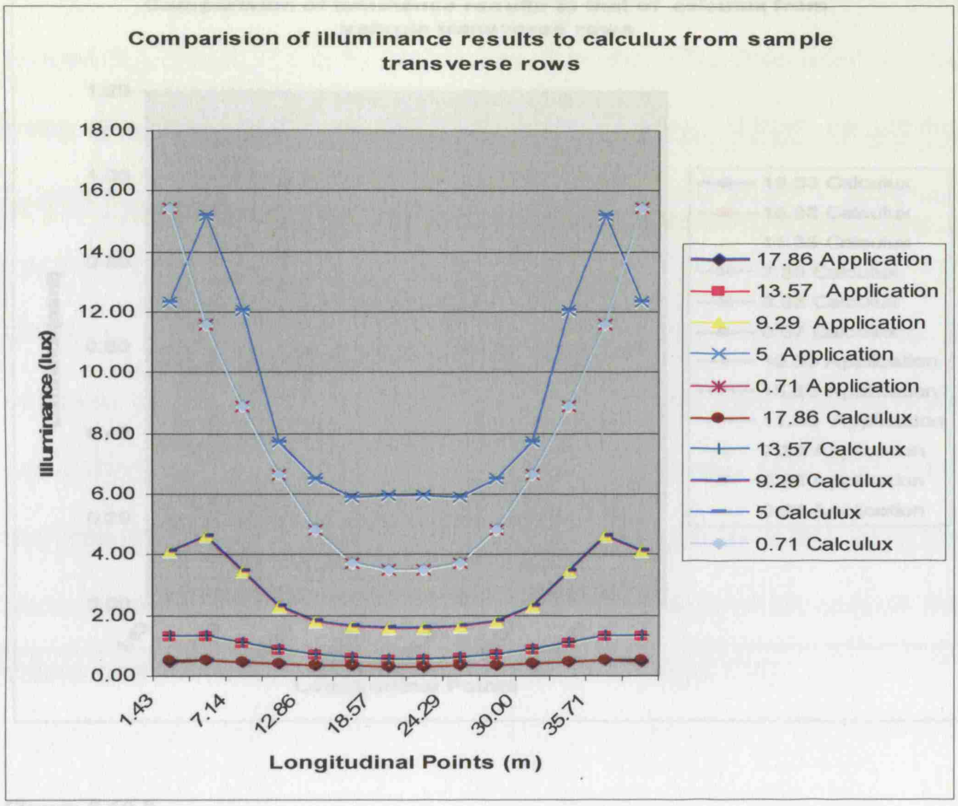



***Figure 3.10.2***

*Comparative chart showing total illuminance results of application to results of Calculux*

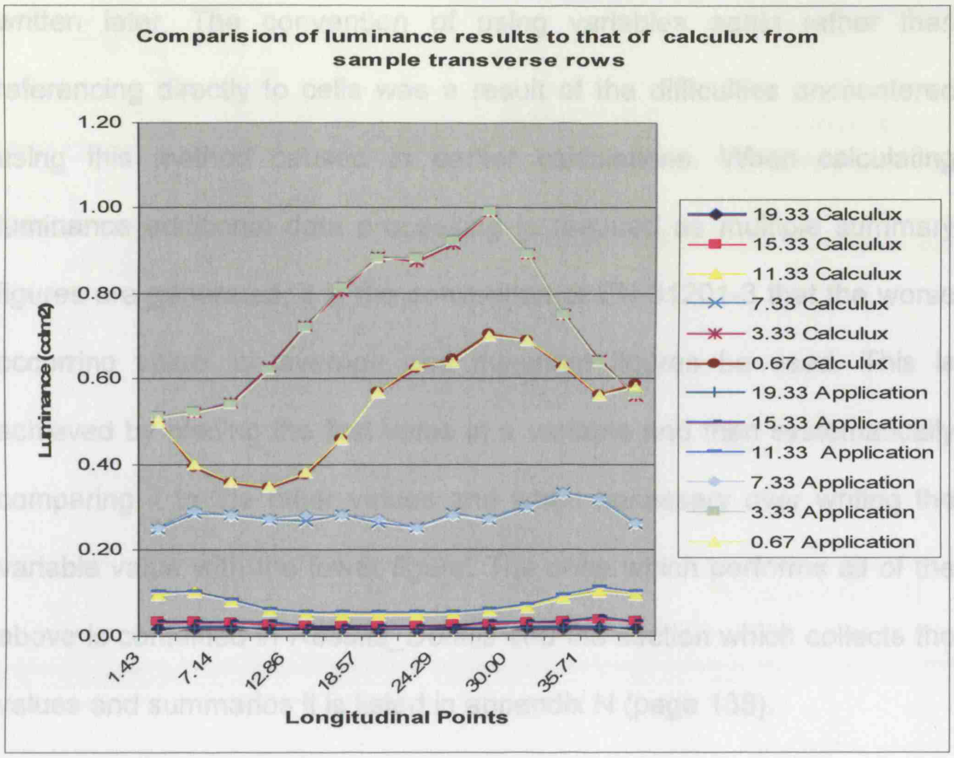



***Figure 3.10.3***

*Comparative chart showing total luminance results of application to result of Calculux*

Once tables of total values are generated the *Results_Details* routine then analyses this information in order to produce summary figures for presentation. This is be achieved in part by Excel's inbuilt functions *Min, Max and Average* to return average, minimum and maximum figures of a selected area. For this reason the grid values are place one cell away from the *distance* row and column is shown in figure 3.10.3, so that the *select.region* command can be used to select the desired area and return the desired value. Values for the average, minimum, maximum and minimum/average are stored in the variables Lum_Ave, Lum_Min, Lum_Max, Lum_Uo and Ul; whilst written into the *Results Details* spreadsheet they are also shown in the *Results Summary* worksheet written later. The convention of using variables again rather than referencing directly to cells was a result of the difficulties encountered using this method caused in earlier calculations. When calculating luminance additional data processing is required as multiple summary figures are generated, it is the convention of EN 31201-3 that the worse occurring value for average and minimum figures be used. This is achieved by placing the first value in a variable and then systematically comparing it to the other values and when necessary over writing the variable value with the lower figure. The code which performs all of the above is contained in *Results_Details* and the section which collects the values and summaries it is listed in appendix N (page 138).

**RESULTS FOR OBSERVER 1**

| Point Locations | 1.50 | 4.50 | 7.50 | 10.50 | 13.50 | 16.50 | 19.50 | 22.50 | 25.50 | 28.50 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11.33 | 1.47 | 1.61 | 1.51 | 1.49 | 1.64 | 1.70 | 1.78 | 1.82 | 1.72 | 1.54 | | |
| 10.00 | 1.82 | 2.00 | 2.03 | 1.84 | 1.89 | 1.93 | 1.98 | 2.17 | 2.04 | 1.86 | UI | 0.84 |
| 8.67 | 1.88 | 2.10 | 2.18 | 1.96 | 1.93 | 1.91 | 1.91 | 2.00 | 2.01 | 1.89 | | |
| 7.33 | 1.83 | 1.96 | 2.07 | 1.94 | 1.91 | 1.86 | 1.87 | 1.93 | 1.84 | 1.86 | | |
| 6.00 | 1.87 | 1.92 | 2.06 | 1.94 | 1.94 | 1.87 | 1.92 | 1.90 | 1.90 | 1.86 | | |
| 4.67 | 2.10 | 2.24 | 2.37 | 2.24 | 2.12 | 2.07 | 2.11 | 2.16 | 2.03 | 2.01 | | |
| 3.33 | 2.28 | 2.46 | 2.69 | 2.68 | 2.30 | 2.23 | 2.41 | 2.40 | 2.14 | 2.16 | | |
| 2.00 | 1.94 | 2.13 | 2.41 | 2.31 | 2.06 | 1.99 | 2.20 | 2.09 | 1.89 | 1.90 | | |
| 0.67 | 1.35 | 1.36 | 1.48 | 1.40 | 1.44 | 1.39 | 1.63 | 1.41 | 1.34 | 1.48 | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| AVERAGE LUMINANCE | 1.93 | MINIMUM LUMINANCE | 1.34 | MAXIMUM LUMINANCE | 2.69 | MINIMUM AVERAGE | 0.69 |

***FIGURE 3.10.3***

*Sample data range laid out for easy select region selection*

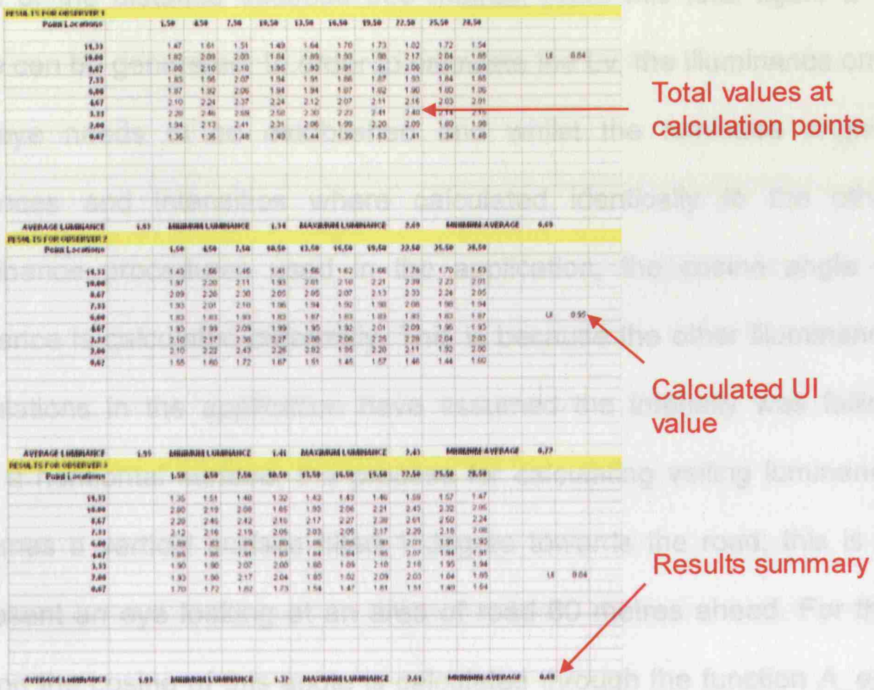



*FIGURE 3.10.4*

*The layout of accumulated data in the Result Details worksheet.*

## 3.11 THRESHOLD INCREMENT

When providing luminance values an additional calculation is required being the threshold increment (TI); this is influenced by the average luminance, the installation type and the luminaire's intensity between 70 and 90 degrees. This was not calculated in the *Luminaire* worksheets as only one point per observer needs to be considered and the quantity of luminaires required to be considered for this calculation will most likely be different. The process requires an observer positioned in each lane so that the veiling luminance (Lv) from each luminaire can be systematically calculated until the contribution is less than 2% of the total value or the distance exceeds 500 metres. From this total figure a TI value can be generated. In order to calculate the Lv, the illuminance onto the eye needs to be established and whilst the luminaire angles, distances and intensities where calculated identically to the other illuminance procedures used in the application, the cosine angle of incidence is calculated differently. This is because the other illuminance calculations in the application have assumed the intensity was falling onto a horizontal surface, the process for calculating veiling luminance assumes a vertical surface tilted 1 degree towards the road; this is to represent an eye looking at an area of road 60 metres ahead. For this reason the cosine of this angle is calculated through the function *A_eye* which generates its value based on the relative locations of the observer, the viewing location and the luminaire.

As the total number of columns is unknown the application generates the contributions of the closest unused luminaire until a threshold is reached. Its method is to list the details required for the calculation process in the same column and move to the next column to repeat the procedure for the next luminaire as shown in figure 3.11.1. When the Lv value for the particular column is established, it is stored in the variable *Temp_Lv_Left* or *Temp_Lv_Right* depending on which road side the column was on, this is then compared as a percentage to the total figure. If the value is greater than 2% the value is added to the variable *Lv_Total* and if not the value is rejected. The details are still displayed in the sheet in order show why the value was rejected but for clarity "Rejected" is written under the value to show it has not been included in the total. This process will not stop until both rows of the installation are rejected to avoid a situation where the process will stop due to a low contribution from one orientation without checking the other. This is achieved through a statement checking for both *Temp_Lv_Left* and *Temp_Lv_Right* to be low. When the threshold is reach the process stops as *Lv_total* contains the total figure and can be used to calculate the TI in the results summary. The code for this section is contained in module *TI* and the process of adding luminaires and calculating and comparing their contribution is listed in appendix P (page 142).

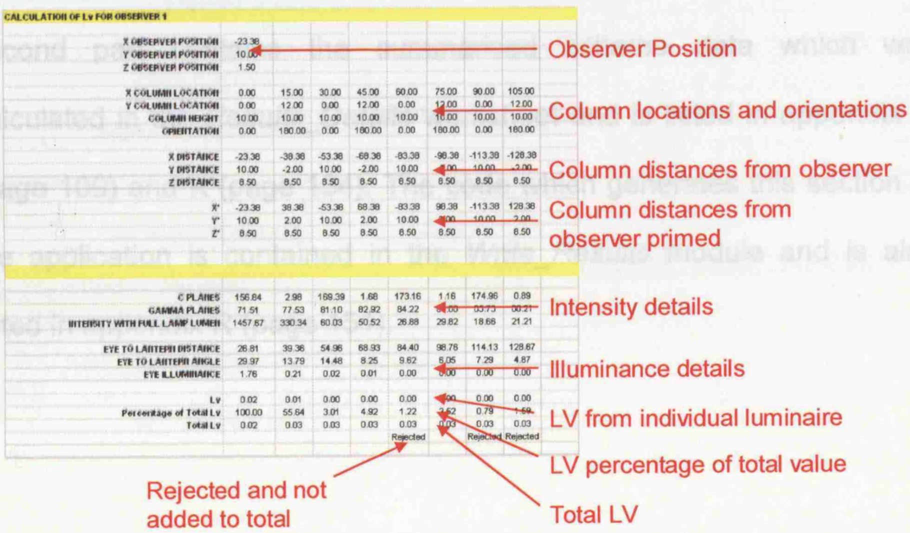



***FIGURE 3.11.1***

*Presentation of the calculated data for Lv*

## 3.12 RESULTS SUMMARY

The results summary displays the scheme specification and the summarised values of the *Results_Details* worksheet as defined by EN31201-3. The purpose of this worksheet is to provide a simple page which clearly shows the results of all the above processes along side the specification in a manner similar to other lighting packages for easily comparison. The *Results_Summary* page of a sample session is shown in figure 3.12.1. The code of this page is simple as no calculations are performed, it simply writes the collected variables into the worksheet. The first data displayed is that of the scheme specification; all this data was written into variables at the beginning of the process when the *Scheme_Editor* form was completed. The code showing the process of assigning these variables values is listed in appendix E (page 97). The second part contains the summarised scheme data which was calculated in the *Results_Details* worksheet and is listed in appendix G (page 109) and K (page 134). The code which generates this section of the application is contained in the *Write_Results* module and is also listed in appendix R (page 151).

# Lighting 'N' Design Application (LiNDA)

**CALCULATION OPTIONS**

| Illuminance Calculation | Luminance Calculation | Get R-Table | Load an Elumdat File |
|---|---|---|---|

**SCHEME DETAILS**

| | | |
|---|---|---|
| INTERPOLATION OF PHOTOMETRY | QUADRATIC | |
| CALCULATION TYPE | LUMINANCE | |
| DRIVING SIDE | LEFT | |
| INSTALATION | SINGLE | |
| NUMBER OF LANES | 3.00 | |
| ROAD SURFACE | Asphalt CIE C2 | |
| Q0 of Table | 0.07 | |
| ROAD WIDTH | 10.00 | metres |
| COLUMN SPACINGS | 30.00 | metres |
| COLUMN HEIGHT | 7.00 | metres |
| LANTERN TILT | 0.00 | degrees |
| OVERHANG | 0.00 | metres |
| LUMINAIRE NAME | SGS201 TP FG P5 | |
| LAMP | SON-TPP100W | |
| LAMP LUMENS | 10700.00 | lumens |
| NUMBER OF LAMPS | 1.00 | |
| MAINTAINENCE FACTOR | 1.00 | |

**SCHEME STATISTICS**

| | | |
|---|---|---|
| AVERAGE LUMINANCE | 1.14 | cdm2 |
| OVERALL UNIFORMITY (Uo) | 0.10 | |
| LONGITUDINAL UNIFORMITY (Ul) | 0.56 | |
| THESHOLD INCREMENT (TI) | 17.28 | % |
| Surround Ratio (SR) | 0.36 | |

***FIGURE 3.12.1***

*Results Summary page*

## 4. RESULTS

The aim of this project was to create a system for checking the accuracy of road lighting software in accordance with EN 13201-3. This was successfully addressed by developing an application using these procedures capable of simulating the same situations as other software available, but with its intermediate calculations exposed for verification. Feed back from the checking bodies related predominantly to usability issues regarding the userforms in particular. A major error was noted in the program's calculation of longitudal uniformity as Ul for all lanes were being considered when in actual fact only the observers lane is of interest. This has been resolved and is the only major error identified to date although a more detailed analysis is expected.

The process of following the document so closely has also proven an excellent method of scrutinising is clarity of the procedures and accuracy of its mathematic formulas. The general format is very clear and the procedures are understandable and applicable to specific calculations. Difficulty was found however, locating the definitions of luminance quality figures being *average, overall uniformity* and *longitudal uniformity* due to an illogical placement with in the standard. A reader would expect this information to be located in the *Calculation of quality characteristics*, but this is actually written in *7.1.5 Position of observer* as shown in figure 4.1.1. This definition was only found by systematically checking the entire document.

**7.1.5 Position of observer**

For luminance calculations the observer's eye is 1,5 m above the road level.

In the transverse direction the observer shall be positioned in the centre of each lane in turn. Average luminance (see 8.2), overall uniformity of luminance (see 8.3) and threshold increment (see 8.5) shall be calculated for the entire carriageway for each position of the observer. Longitudinal uniformity of luminance (see 8.4) shall be calculated for each centre-line. The operative values of average luminance, overall uniformity of luminance, and longitudinal uniformity of luminance shall be the lowest in each case; the operative value of threshold increment shall be the highest value.

***FIGURE 4.1.1***

*Definition of summery values as stated in EN 13021-3*

More crucially an error was found in the mathematical formula for the calculation of H primed as shown in figure 4.1.1; *sin* is used incorrectly in the formula as the actual required value is *cos.* This error was identified as a result of checking all the illuminance program code after large discrepancies where found between the values generated by the application and Calculux. This error was verified by cross referencing the formula in EN 13021-3 with that of CIE140 as shown in figure 4.1.3, and *Lighting Engineering: Applied Calculations*[18]. This has been communicated to Ron Simons who has further verified the error and British Standards have been notified.

[18] Simons, R.H. & Bean, A.R. *Lighting Engineering: Applied Calculations.* London(2000)

**6.4 Calculation of $C$ and $\gamma$**

These can be determined in four stages:

1) Substitution of $v$, $\psi$, $\delta$, $x$ and $y$ in the equations:

$$x' = x(\cos v \ \cos\psi - \sin v \ \sin\delta \ \sin\psi) + y(\sin v \ \cos\psi + \cos v \ \sin\delta \ \sin\psi) + H \ \cos\delta \ \sin\psi \qquad (21)$$

$$y' = -x \ \sin v \ \cos\delta + y \ \cos v \ \cos\delta - H \ \sin\delta \qquad (22)$$

$$H' = -x(\sin v \ \sin\delta \ \cos\psi + \cos v \ \sin\psi) - y(\sin v \ \sin\psi - \cos v \ \sin\delta \ \cos\psi) + H \ \cos\delta \ \sin\psi \qquad (23)$$

where:

$x$ and $y$ are the longitudinal and transverse distances between the calculation point and the nadir of the luminaire in Figure 6

$H$ is the height of the luminaire above the calculation point

***FIGURE 4.1.2***

*Formula for calculation of H' stated in EN13021-3*

**6.4 Calculation of $C$ and $\gamma$**

These can be determined in four stages:

1) Substitution of $v$, $\psi$, $\delta$, $x$ and $y$ in the equations:

$$x' = x(\cos v \ \cos\psi - \sin v \ \sin\delta \ \sin\psi) + y(\sin v \ \cos\psi - \cos v \ \sin\delta \ \sin\psi) + H \ \cos\delta \ \sin\psi \qquad (21)$$

$$y' = -x \ \sin v \ \cos\delta + y \ \cos v \ \cos\delta - H \ \sin\delta \qquad (22)$$

$$H' = -x(\sin v \ \sin\delta \ \cos\psi + \cos v \ \sin\psi) - y(\sin v \ \sin\psi - \cos v \ \sin\delta \ \cos\psi) + H \ \cos\delta \ \cos\psi \qquad (23)$$

where:

$x$ and $y$ are the longitudinal and transverse distances between the calculation point and the nadir of the luminaire in Figure 6

$H$ is the height of the luminaire above the calculation point

***FIGURE 4.1.3***

*Formula for calculation of H' stated in CIE140*

As EN31201-3 has defined formulas for the linear and quadratic interpolation of intensities, their consistency to one another was evaluated by calculating identical schemes with each method. Comparison between the two showed similar results in all areas with the largest fluctuation occurring in TI values as shown in figures 4.1.4 and 4.1.5. This difference becomes more pronounced as the column height is increased but even at a column height of twelve metres being the maximum commonly used on UK roads, the difference was 0.1% between the two.

| | | | | |
|---|---|---|---|---|
| **SCHEME DETAILS** | | | | |
| INTERPOLATION OF PHOTOMETRY | QUADRATIC | | LINEAR | |
| CALCULATION TYPE | LUMINANCE | | LUMINANCE | |
| DRIVING SIDE | LEFT | | LEFT | |
| INSTALLATION | STAGGERED | | STAGGERED | |
| NUMBER OF LANES | 3.00 | | 3.00 | |
| ROAD SURFACE | Asphalt CIE C2 | | Asphalt CIE C2 | |
| Q0 of Table | 0.07 | | 0.07 | |
| ROAD WIDTH | 12.00 | metres | 12.00 | metres |
| COLUMN SPACINGS | 40.00 | metres | 40.00 | metres |
| COLUMN HEIGHT | 12.00 | metres | 12.00 | metres |
| LANTERN TILT | 3.00 | degrees | 3.00 | degrees |
| OVERHANG | 0.50 | metres | 0.50 | metres |
| LUMINAIRE NAME | SGS252 FG OR P5X | | SGS252 FG OR P5X | |
| LAMP | CDO-TT70W | | CDO-TT70W | |
| LAMP LUMENS | 6300.00 | lumens | 6300.00 | lumens |
| NUMBER OF LAMPS | 1.00 | | 1.00 | |
| MAINTAINENCE FACTOR | 0.80 | | 0.80 | |
| **SCHEME STATISTICS** | | | | |
| AVERAGE LUMINANCE | 0.61 | cdm2 | 0.61 | cdm2 |
| OVERALL UNIFORMITY (Uo) | 0.57 | | 0.57 | |
| LONGITUDINAL UNIFORMITY (Ul) | 0.81 | | 0.80 | |
| THESHOLD INCREMENT (TI) | 4.09 | % | 4.07 | % |
| Surround Ratio (SR) | 0.42 | | 0.43 | |

*FIGURE 4.1.4*

*Results Summary of sample calculation using linear interpolation*

| | | | | | |
|---|---|---|---|---|---|
| **SCHEME DETAILS** | | | | | |
| INTERPOLATION OF PHOTOMETRY | QUADRATIC | | | LINEAR | |
| CALCULATION TYPE | LUMINANCE | | | LUMINANCE | |
| DRIVING SIDE | LEFT | | | LEFT | |
| INSTALLATION | STAGGERED | | | STAGGERED | |
| NUMBER OF LANES | 3.00 | | | 3.00 | |
| ROAD SURFACE | Asphalt CIE C2 | | | Asphalt CIE C2 | |
| Q0 of Table | 0.07 | | | 0.07 | |
| ROAD WIDTH | 12.00 | metres | | 12.00 | metres |
| COLUMN SPACINGS | 40.00 | metres | | 40.00 | metres |
| COLUMN HEIGHT | 6.00 | metres | | 6.00 | metres |
| LANTERN TILT | 3.00 | degrees | | 3.00 | degrees |
| OVERHANG | 0.50 | metres | | 0.50 | metres |
| LUMINAIRE NAME | SGS252 FG OR P5X | | | SGS252 FG OR P5X | |
| LAMP | CDO-TT70W | | | CDO-TT70W | |
| LAMP LUMENS | 6300.00 | lumens | | 6300.00 | lumens |
| NUMBER OF LAMPS | 1.00 | | | 1.00 | |
| MAINTAINENCE FACTOR | 0.80 | | | 0.80 | |
| **SCHEME STATISTICS** | | | | | |
| AVERAGE LUMINANCE | 0.74 | cdm2 | | 0.73 | cdm2 |
| OVERALL UNIFORMITY (Uo) | 0.21 | | | 0.21 | |
| LONGITUDINAL UNIFORMITY (Ul) | 0.13 | | | 0.14 | |
| THESHOLD INCREMENT (TI) | 12.33 | % | | 12.43 | % |
| Surround Ratio (SR) | 0.33 | | | 0.34 | |

***FIGURE 4.1.5***

*Results Summary of sample calculation using quadratic interpolation*

Further study of the quadratic routine revealed irregularities when interpolating values at intermediate angles, this is due to the definition of the angles of reference used in the formula. The procedure dictates that the two tabular angles adjacent to the angle for interpolation be used, if the average of these is smaller than the angle to be interpolated, the third angle used is the next lower on the tabular value, if the value is larger then the next higher value is used. This means that values interpolated across the centre point at measured angles will contain a fluctuation due to the reference points changing as shown in figure 5.1.6. In the cases tested this variation was only in the region of 2 candelas per 1000 lumens but it was also found that other programs tested did not suffer with this issue.

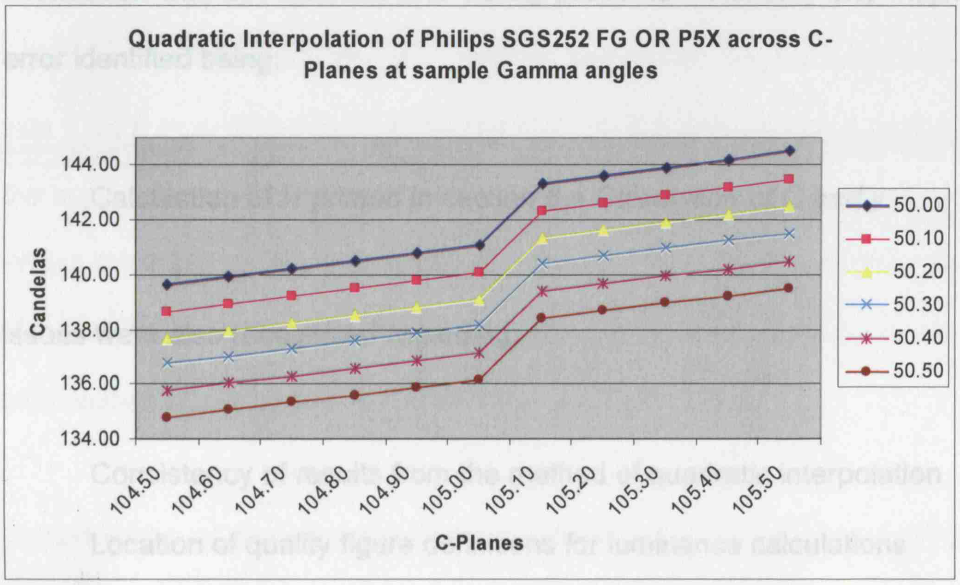



*FIGURE 4.1.4*

*Chart showing quadratic interpolation across centre of measured angles*

## 5. CONCLUSION

The scope of EN 13201-3 is to define and describe the conventions and mathematical procedures used in the calculation of the photometric performance of the road lighting installations in order to generate quality characteristics as specified in EN13021-2. Therefore an evaluation of the success of this scope can be suggested by considering the feasibility of this project by referring purely to this resource. All the necessary information was documented and clearly presented with only one major error identified being:

- Calculation of *H primed* in section *6.4 Calculation of C and y*

Issues were also recognized regarding:

- Consistency of results from the method of quadratic interpolation
- Location of quality figure definitions for luminance calculations

Disregarding the formula error, this project is proof that EN 13021-3 definitely contains the necessary information to calculate the photometric qualities of a road lighting installation and even expand this concept into an application.

## 6. FURTHER RESEARCH

This project will be further developed by firstly verifying its functions under deeper scrutiny to attain a far greater degree of credibility and secondly, making the verified application available to the lighting community for use in the industry. Whilst the checking bodies agreed that the program seems to be generating results in accordance with the procedures stated in EN13201-3, a deeper analysis into its intermediate functions should be carried out if the application is to attain any level of bench mark status. As the only certified calculation engine developed was *LUCY* and *STAN* to accompany CIE 30.2 a comparison between the application and LUCY could be undergone to measure its accuracy. Whilst the mathematical procedures between the two are likely to be identical, for this exercise to have any meaning these processes and the conventions *LUCY* and *STAN* need to be first understood.

Consideration of the usability of the product to the lighting community was noted in the project objectives and has been addressed by using the *Microsoft Excel* platform. In order to make the application accessible web space from where the application could downloaded is being sought; opportunity might be available to list it on the *Institute of Lighting Engineers (ILE)*[19] website but this would of cause be pending the application's further verification. When the program is fully verified and accessible, the obvious opportunity for further study is to actually use it in a comparative study of the road lighting software available. As a result

---

[19] http://www.ile.org.uk/

software's conformance with the conventions of EN31201-3 can be measured and publicised to the lighting community.

## 7. BIBLIOGRAPHY


### BOOKS

Bommel, W.J.M. and Boer, J.B., *Road Lighting*. Kluwer Teechnische Boeken (1980)

*CEN/TR 13201-1*. BSI (2004)

*EN 13201-2*. BSI (2004)

*EN 13201-3*. BSI (2004)

*EN 13201-4*. BSI (2004)

*CIE 140-2000*. CIE Central Bureau (2000)

*CIE 30-2*. CIE Central Bureau (1982)

*Philips Lighting Manual 5th Edition*. Philip Lighting B.V. (1993)

Pritchard, D. C., *Lighting 6th Edition*. Longman (1999)

Simons, R.H. & Bean, A.R. *Lighting Engineering: Applied Calculations*. London (2000)

### LECTURES

Raynham, P., *L06 - road lighting supporting notes*. Lecturing at University College London (2006)

### WEBSITES

http://www.lighting.philips.com/gl_en/tools_downloads/calculuxdialux/downloads.php?main=global&parent=4390&id=gl_en_calculuxdialux&lang=en (2006)

http://www.ile.org.uk/ (2006)

# APPENDIX

## APPENDIX CONTENTS

## APPENDIX A

Section of file browser which collect folders and displays them *Directory* list

---

```
'--------------------------------CREATES A LIST OF DIRECTORIES BASED ON THE PATH Photometry'------------------
Sub Updat_Directory_List()
Dim Found As String
Dim Dirs(500) As String 'CAN HOLD MAX OF 500 DIRECTORIES
Dim Found_List(600) As String ''CAN HOLD MAX OF 600 ANY FILES
Dim i As Integer
Dim LastFound As Integer
Dim j As Integer
Dim k As Integer
i = 1
MyDir = Worksheets("data").Cells(2, 1)
FilePath = MyDir
'--------------------------------CREATES A LIST ALL FOUND ITEMS IN THE PATH FOLDER'--------------------------------
On Error GoTo Error_Message 'catches error if file being looked in does not exist
Found = Dir(MyDir & "*", vbDirectory)
i = 1
If Found <> "" Then
   Found_List(1) = Found
   i = 2
End If
Do Until Found = ""
   Found = Dir
```

```
   If Found <> "" Then
      Found_List(i) = Found
      i = i + 1
      If i = 600 Then
         MsgBox "Too many files to read!"
         Worksheets("data").Cells(1, 1) = ""
         Worksheets("data").Cells(2, 1) = ""
         Drive_Name_AfterUpdate
   Exit Sub  'FROM FREEZING
   End If
   End If
Loop
LastFound = i - 1
'---------------------------------CREATES A NEW LIST FROM THE PREVIOUS WITH ONLY THE DIRECT
-------------------
j = 1
For i = 1 To LastFound
   If Check_If_Directory(Found_List(i)) Then
      Dirs(j) = Found_List(i)
      j = j + 1
   End If
Next i
'---------------------------------SORTS THE LIST ALPHABETICALLY---------------------------------
LastFound = j - 1
For i = 1 To LastFound
   For j = 1 To LastFound - i
      If UCase(Dirs(j)) > UCase(Dirs(j + 1)) Then
         Found = Dirs(j)
```

```
          Dirs(j) = Dirs(j + 1)
          Dirs(j + 1) = Found
       End If
   Next j
Next i
'---------------------------------CLEARS THE EXISTING FORM LIST AND REWRITES IT WITH THE NEW
-
Directory_List.Clear
Directory_List.AddItem ".." 'SYMBOL TO GT TO THE PARENT DIRECTORY WHEN DOUBLE CLICKE
For i = 1 To LastFound
   Directory_List.AddItem Dirs(i)
Next i
 First_File_Lines.Text = ""
 Update_File_List
Exit Sub
Error_Message:
MsgBox "Not a Valid Drive"
Worksheets("data").Cells(2, 1) = ""
Worksheets("data").Cells(1, 1) = ""
MyDir = ""
Drive_Name.Text = ""
Drive_Name = ""
Updat_Directory_List
End Sub
```

**APPENDIX B**

Section of I-table write which establishes luminaire symmetry type

```
'--------------------------------writes in the I-Table angles--------------------------------
Worksheets("Photometry").Cells(50, 1).Value = "I-Table"
Num_C_Angles = Worksheets("Photometry").Cells(8, 6).Value
Num_Gamma_Angles = Worksheets("Photometry").Cells(10, 6).Value
ReDim C_angles(Num_C_Angles)
ReDim Gamma_Angles(Num_Gamma_Angles)
For i = 1 To Num_C_Angles
   Input #1, C_angles(i)
Next i
For i = 1 To Num_Gamma_Angles
   Input #1, Gamma_Angles(i)
Next i
'--------------------------------sorts the angles based on symmetry indicator--------------------------------
Select Case Sym_Indicator
Case Is = 0
Case Is = 1
      Num_C_Angles = 1
Case Is = 2
   Num_C_Angles = (Num_C_Angles / 2) + 1
Case Is = 3
   Num_C_Angles = (Num_C_Angles / 2) + 1
   ReDim C_Angles_Temp(Num_C_Angles)
   Degrees90 = 1
```

```
  Do Until C_angles(Degrees90) = 90
     If C_angles(Degrees90) > 90 Then GoTo Error_Message
     Degrees90 = Degrees90 + 1
  Loop
  For i = 1 To Num_C_Angles
     C_Angles_Temp(i) = C_angles(i + Degrees90 - 1)
  Next i
  If C_Angles_Temp(Num_C_Angles) <> 270 Then GoTo Error_Message
  ReDim C_angles(Num_C_Angles)
  For i = 1 To Num_C_Angles
     C_angles(i) = C_Angles_Temp(i)
  Next i
Case Is = 4
  Num_C_Angles = (Num_C_Angles / 4) + 1
 End Select
'---------------------------------writes the intensities into an array---------------------------------
ReDim I_Table(Num_Gamma_Angles, Num_C_Angles)
If Sym_Indicator = 3 Then
For k = Num_C_Angles To 1 Step -1
  For j = 1 To Num_Gamma_Angles
     Input #1, I_Table(j, k)
  Next j
Next k
Else
For k = 1 To Num_C_Angles
  For j = 1 To Num_Gamma_Angles
     Input #1, I_Table(j, k)
  Next j
```

```
Next k
End If
Close 1
'--------------------------------writes the angles and intensities array into spread sheet----------------------------
For i = 1 To Num_Gamma_Angles
Worksheets("Photometry").Cells(i + 50, 1).Value = Gamma_Angles(i)
Next i
For i = 1 To Num_C_Angles
Worksheets("Photometry").Cells(50, i + 1).Value = C_angles(i)
Next i
For k = 1 To Num_C_Angles
   For j = 1 To Num_Gamma_Angles
      Worksheets("Photometry").Cells(j + 50, k + 1).Value = I_Table(j, k)
   Next j
Next k
Exit Sub
Error_Message:
   Close 1
   MsgBox "Unexpected Set of C-Planes"
End Sub
```

**APPENDIX C**

Routines for linear and quadratic interpolation

---

```
      If Cub_Inten = True Then
         Cubic_Intensity = Cubic_I3(Used_Azimuth, points(1), points(2), points(3), values(1), values(2),
      Else
         Cubic_Intensity = linear_interpolation(Used_Azimuth, points(1), points(2), values(1), values(2))
      End If
End Select
```

---

```
Function Cubic_I3(c, CM, CM1, CM2, VM, VM1, VM2) As Double
Dim PartProd As Double

PartProd = VM * Cubic_K1(c, CM, CM1, CM2)
PartProd = PartProd + VM1 * Cubic_K2(c, CM, CM1, CM2)
PartProd = PartProd + VM2 * Cubic_K3(c, CM, CM1, CM2)
Cubic_I3 = PartProd
End Function
```

---

```
Function Cubic_K3(c, CM, CM1, CM2) As Double
Dim PartProd As Double
PartProd = (c - CM) * (c - CM1)
PartProd = PartProd / ((CM2 - CM) * (CM2 - CM1))
Cubic_K3 = PartProd
End Function
```

```
Function Cubic_K2(c, CM, CM1, CM2) As Double
Dim PartProd As Double
PartProd = (c - CM) * (c - CM2)
PartProd = PartProd / ((CM1 - CM) * (CM1 - CM2))
Cubic_K2 = PartProd
End Function
```

---

```
Function Cubic_K1(c, CM, CM1, CM2) As Double
Dim PartProd As Double
PartProd = (c - CM1) * (c - CM2)
PartProd = PartProd / ((CM - CM1) * (CM - CM2))
Cubic_K1 = PartProd
End Function
```

---

```
Function linear_interpolation(c_lin, lower_point_lin, higher_point_lin, lower_value_lin, higher_value_lin) A
linear_interpolation = lower_value_lin + (higher_value_lin - lower_value_lin) * (c_lin - lower_point_lin) / (h
lower_point_lin)
End Function
```

## APPENDIX D.1

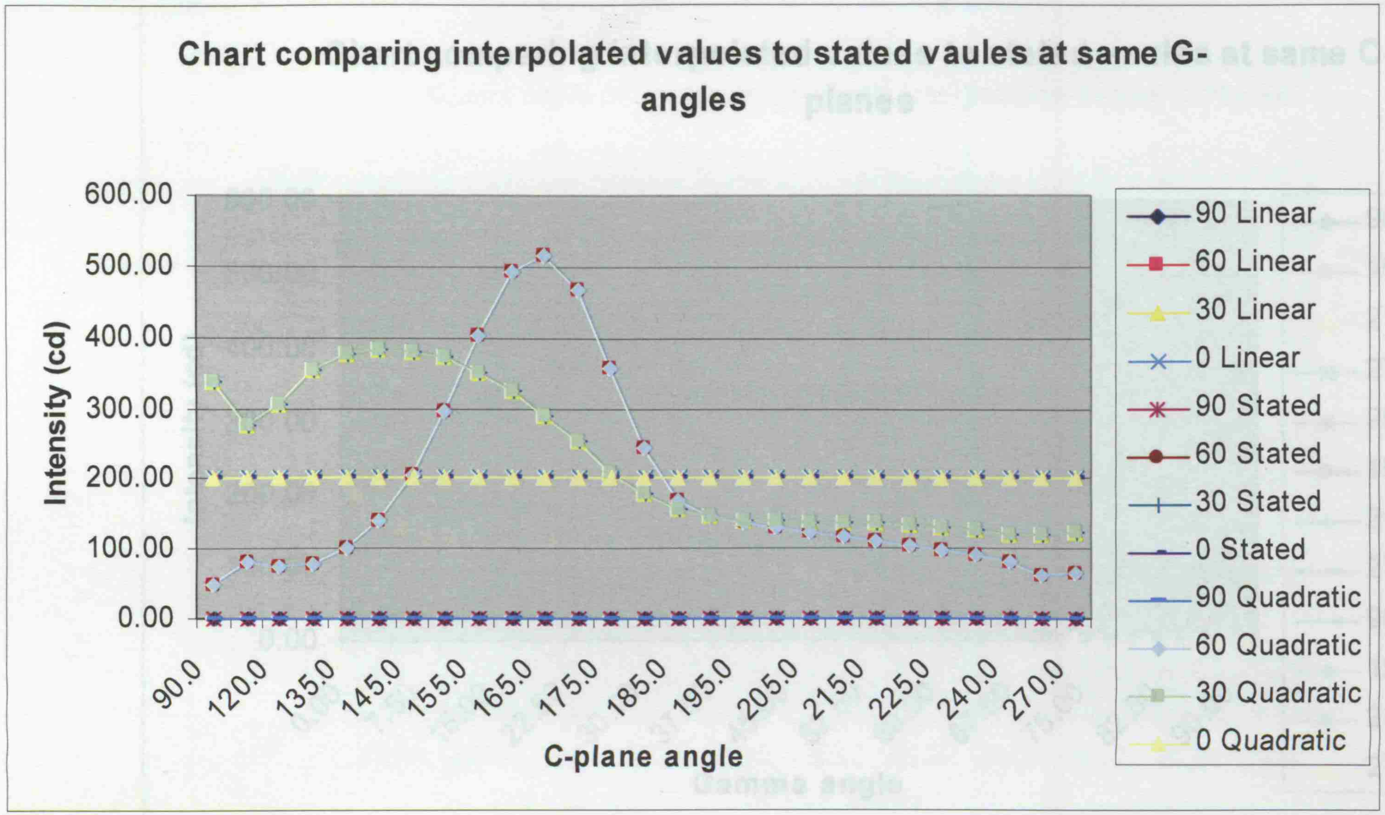


Chart comparing interpolated values to stated vaules at same G-angles
Intensity (cd)
600.00
500.00
400.00
300.00
200.00
100.00
0.00
90.0
120.0
135.0
145.0
155.0
165.0
175.0
185.0
195.0
205.0
215.0
225.0
240.0
270.0
C-plane angle
90 Linear
60 Linear
30 Linear
0 Linear
90 Stated
60 Stated
30 Stated
0 Stated
90 Quadratic
60 Quadratic
30 Quadratic
0 Quadratic

APPENDIX D.2

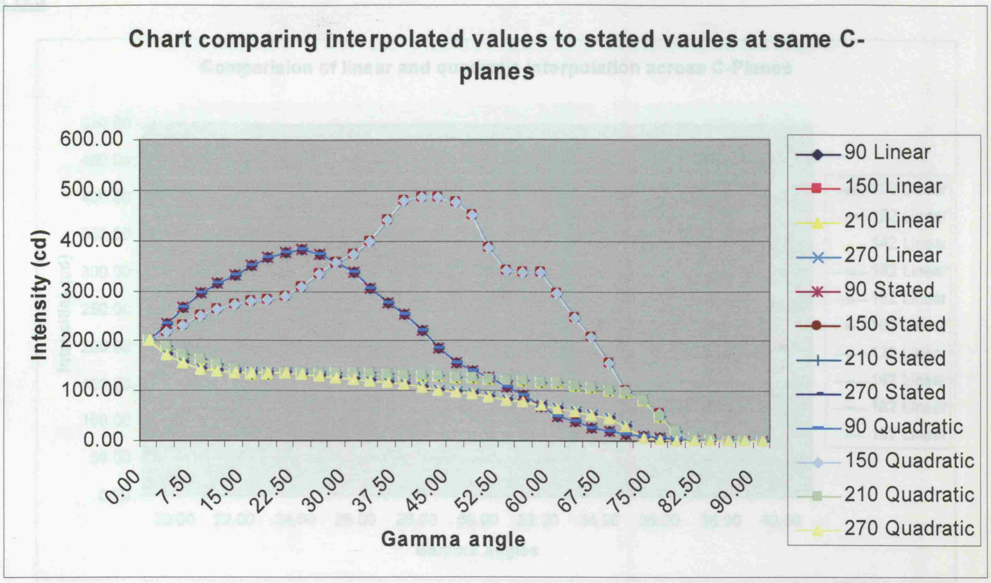
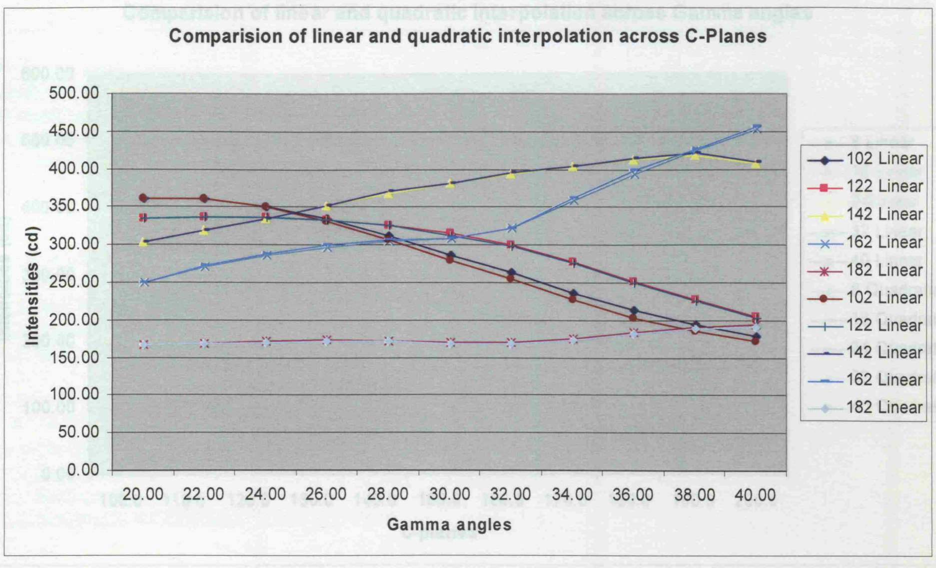


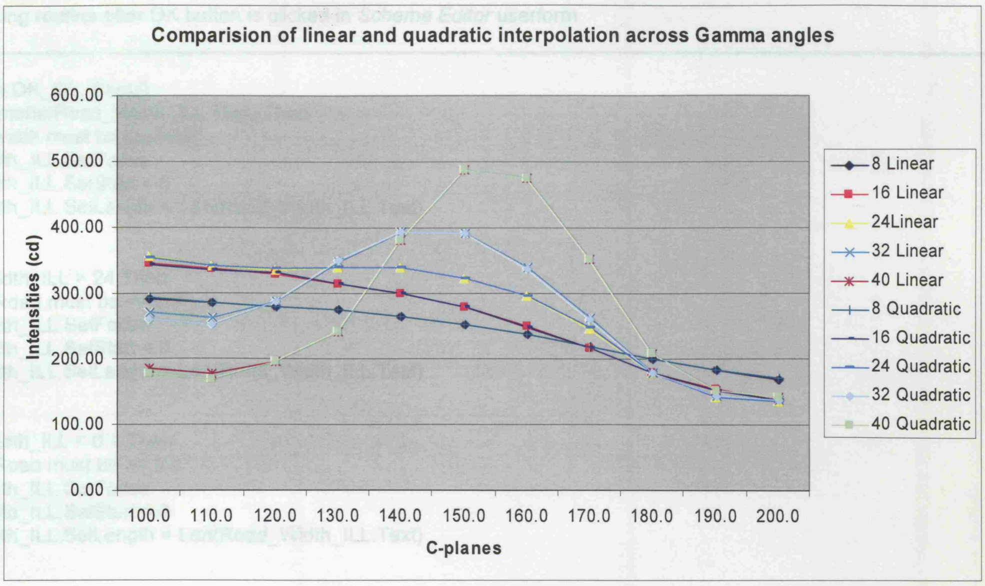
Chart comparing interpolated values to stated vaules at same C-planes
600.00
500.00
400.00
300.00
200.00
100.00
0.00
Intensity (cd)
0.00
7.50
15.00
22.50
30.00
37.50
45.00
52.50
60.00
67.50
75.00
82.50
90.00
Gamma angle
90 Linear
150 Linear
210 Linear
270 Linear
90 Stated
150 Stated
210 Stated
270 Stated
90 Quadratic
150 Quadratic
210 Quadratic
270 Quadratic

APPENDIX D.3



Comparision of linear and quadratic interpolation across C-Planes
Intensities (cd)
500.00
450.00
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
20.00
22.00
24.00
26.00
28.00
30.00
32.00
34.00
36.00
38.00
40.00
Gamma angles
102 Linear
122 Linear
142 Linear
162 Linear
182 Linear
102 Linear
122 Linear
142 Linear
162 Linear
182 Linear

## APPENDIX D.4



Comparision of linear and quadratic interpolation across Gamma angles
Intensities (cd)
600.00
500.00
400.00
300.00
200.00
100.00
0.00
100.0
110.0
120.0
130.0
140.0
150.0
160.0
170.0
180.0
190.0
200.0
C-planes
8 Linear
16 Linear
24Linear
32 Linear
40 Linear
8 Quadratic
16 Quadratic
24 Quadratic
32 Quadratic
40 Quadratic

## APPENDIX E

Data checking routine after OK button is clicked in *Scheme Editor* userform

---

```
Private Sub OK_ILL_Click()
If Not IsNumeric(Road_Width_ILL.Text) Then
 MsgBox "width must be numeric"
 Road_Width_ILL.SetFocus
 Road_Width_ILL.SelStart = 0
 Road_Width_ILL.SelLength = Len(Road_Width_ILL.Text)
Exit Sub
End If
If Road_Width_ILL > 24 Then
 MsgBox "Road must be <= 24"
 Road_Width_ILL.SetFocus
 Road_Width_ILL.SelStart = 0
 Road_Width_ILL.SelLength = Len(Road_Width_ILL.Text)
Exit Sub
End If
If Road_Width_ILL < 0.1 Then
 MsgBox "Road must be >= 0.1"
 Road_Width_ILL.SetFocus
 Road_Width_ILL.SelStart = 0
 Road_Width_ILL.SelLength = Len(Road_Width_ILL.Text)
Exit Sub
End If
```

```
'-------------------------------------------------------------------------------------------
If Installation_ILL = "TWIN CENTRAL" Then
If Not IsNumeric(Central_Divide_Width.Text) Then
 MsgBox "Central divide width height must be numeric"
 Central_Divide_Width.SetFocus
 Central_Divide_Width.SelStart = 0
 Central_Divide_Width.SelLength = Len(Col_Height_ILL.Text)
Exit Sub
End If
End If
If Installation_ILL = "TWIN CENTRAL" Then
If Not Central_Divide_Width < 0 Then
 MsgBox "Central divide width height must be => 0"
 Central_Divide_Width.SetFocus
 Central_Divide_Width.SelStart = 0
 Central_Divide_Width.SelLength = Len(Col_Height_ILL.Text)
Exit Sub
End If
End If
'-------------------------------------------------------------------------------------------
If Not IsNumeric(Col_Height_ILL.Text) Then
 MsgBox "Column height must be numeric"
 Col_Height_ILL.SetFocus
 Col_Height_ILL.SelStart = 0
 Col_Height_ILL.SelLength = Len(Col_Height_ILL.Text)
Exit Sub
End If
If Col_Height_ILL < 1 Then
```

```
 MsgBox "Column height must >= 1"
 Col_Height_ILL.SetFocus
 Col_Height_ILL.SelStart = 0
 Col_Height_ILL.SelLength = Len(Col_Height_ILL.Text)
Exit Sub
End If
'---------------------------------------------------------------------------------------------------
If Not IsNumeric(Spacing_ILL.Text) Then
 MsgBox "spacing must be numeric"
 Spacing_ILL.SetFocus
 Spacing_ILL.SelStart = 0
 Spacing_ILL.SelLength = Len(Spacing_ILL.Text)
Exit Sub
End If
If Spacing_ILL < 1 Then
 MsgBox "spacing must be >= 1"
 Spacing_ILL.SetFocus
 Spacing_ILL.SelStart = 0
 Spacing_ILL.SelLength = Len(Spacing_ILL.Text)
Exit Sub
End If
If Spacing_ILL > 750 Then
 MsgBox "spacing must be >= 750"
 Spacing_ILL.SetFocus
 Spacing_ILL.SelStart = 0
 Spacing_ILL.SelLength = Len(Spacing_ILL.Text)
Exit Sub
End If
```

```
'------------------------------------------------------------------------------------------------
If Not IsNumeric(Overhang_ILL.Text) Then
 MsgBox "Overhang must be numeric"
 Overhang_ILL.SetFocus
 Overhang_ILL.SelStart = 0
 Overhang_ILL.SelLength = Len(Overhang_ILL.Text)
Exit Sub
End If
'------------------------------------------------------------------------------------------------
If Not IsNumeric(tilt_ill.Text) Then
 MsgBox "tilt must be numeric"
 tilt_ill.SetFocus
 tilt_ill.SelStart = 0
 tilt_ill.SelLength = Len(tilt_ill.Text)
Exit Sub
End If
If tilt < -90 Then
 MsgBox "tilt must be => -90"
 tilt_ill.SetFocus
 tilt_ill.SelStart = 0
 tilt_ill.SelLength = Len(tilt_ill.Text)
Exit Sub
End If
If tilt > 90 Then
 MsgBox "tilt must be => 90"
 tilt_ill.SetFocus
 tilt_ill.SelStart = 0
 tilt_ill.SelLength = Len(tilt_ill.Text)
```

```
Exit Sub
End If
'-----------------------------------------------------------------------------------------------------
If Not IsNumeric(MF_ILL.Text) Then
 MsgBox "Maintainence Factor must be numeric"
 MF_ILL.SetFocus
 MF_ILL.SelStart = 0
 MF_ILL.SelLength = Len(MF_ILL.Text)
Exit Sub
End If
If MF_ILL > 1 Then
 MsgBox "Maintainence Factor must be =< 1"
 MF_ILL.SetFocus
 MF_ILL.SelStart = 0
 MF_ILL.SelLength = Len(MF_ILL.Text)
Exit Sub
End If
If MF_ILL < 0 Then
 MsgBox "Maintainence Factor must be =< 0"
 MF_ILL.SetFocus
 MF_ILL.SelStart = 0
 MF_ILL.SelLength = Len(MF_ILL.Text)
Exit Sub
End If
'-------------------------------------------------------------------------------
'If Install_Ill = False Then
'MsgBox "Select Instalation Type"
'Installation_ILL.SetFocus
```

```
'Exit Sub
'Install_Ill = False
'End If
If Installation_ILL.Selected(0) = True Then
Inst_Type = "SINGLE"
End If
If Installation_ILL.Selected(1) = False Then
Inst_Type = "OPPOSITE"
End If
If Installation_ILL.Selected(2) = False Then
Inst_Type = "STAGGERED"
End If
If Installation_ILL.Selected(3) = False Then
Inst_Type = "TWIN CENTRAL"
End If
If Installation_ILL.Selected(0) = False Then
If Installation_ILL.Selected(1) = False Then
If Installation_ILL.Selected(2) = False Then
If Installation_ILL.Selected(3) = False Then
   MsgBox "Please select installation type"
   Installation_ILL.SetFocus
   Exit Sub
End If
End If
End If
End If
'------------------------------------------------------------------------------
number_of_lamps = Worksheets("Photometry").Cells(31, 6).Value
```

```
lamp_lumens = Worksheets("Photometry").Cells(33, 6).Value
maintainance_factor = MF_ILL
column_height = Col_Height_ILL
Column_Spacings = Spacing_ILL
Inst_Type = Installation_ILL
Pole_Location = Road_Width_ILL
Road_Width = Road_Width_ILL
Road_Over_Hang = Overhang_ILL
tilt = tilt_ill
If Inst_Type = "TWIN CENTRAL" Then
Centre_Width = Central_Divide_Width
End If
If Quadratic_ILL = True Then
   Quad_Inten = True
Else
   Quad_Inten = False
End If
If Left_ILL = True Then
   Left_Drive = True
   Else
   Left_Drive = False
End If
Scheme_Editor_ILL.Hide
Collect_Workbook_Photometry_Illuminance
End Sub
'Private Sub Spacing_ILL_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
'Select Case KeyAscii
'Case Is = 46
```

```
'Case 48 To 57
'Case Else
'   KeyAscii = 7
'End Select
'End Sub
Private Sub UserForm_Activate()
Central_Divide_Width.Visible = False
Central_Label.Visible = False
Central_Label2.Visible = False
Install_Ill = False
 Installation_ILL.Clear
 Installation_ILL.AddItem "SINGLE"
 Installation_ILL.AddItem "OPPOSITE"
 Installation_ILL.AddItem "STAGGERED"
 Installation_ILL.AddItem "TWIN CENTRAL"
Select Case Inst_Type
Case Is = ""
Case Is = "SINGLE"
   Installation_ILL.Selected(0) = True
Case Is = "OPPOSITE"
   Installation_ILL.Selected(1) = True
Case Is = "STAGGERED"
   Installation_ILL.Selected(2) = True
Case Is = "TWIN CENTRAL"
   Installation_ILL.Selected(3) = True
End Select
End Sub
```

**APPENDIX F**

Function to calucation number of calculation points and columns

---

```
Function num_of_long_points(Spacings)
Dim point_space As Double
num_of_long_points = 10
point_space = Spacings / num_of_long_points
If Spacings > 30 Then
Do Until point_space <= 3
num_of_long_points = num_of_long_points + 1
point_space = Spacings / num_of_long_points
Loop
End If
End Function
```

---

```
'---------checks the distance between the transverse calc points-----------
Function num_of_trans_points_Illuminance(Road_Width)
Dim point_space As Double
num_of_trans_points_Illuminance = 3
point_space = Road_Width / num_of_trans_points_Illuminance
Do Until point_space <= 1.5
num_of_trans_points_Illuminance = num_of_trans_points_Illuminance + 1
point_space = Road_Width / num_of_trans_points_Illuminance
Loop
```

```
End Function
Function num_of_trans_points_LUMINANCE(Num_Of_Lanes_Temp)
num_of_trans_points_LUMINANCE = Num_Of_Lanes_Temp * 3
End Function
```

---

```
Function Num_Of_Columns_Illuminance(column_height, Calc_Point_Location, final_calc_point_locatio
Inst_Type) As Integer
Dim Next_Column_Location As Double
'-------------------works out number of required columns ILLUMINANCE---------------------------------
num_of_columns_infront = 0
num_of_columns_behind = 0
include_dist_infront = column_height * 5
include_location_infront = include_dist_infront + final_calc_point_location
include_dist_behind = column_height * 5
include_location_behind = include_dist_behind - (include_dist_behind * 2) + first_calc_point


Next_Column_Location = 0
Do Until Next_Column_Location < include_location_behind
num_of_columns_behind = num_of_columns_behind + 1
Next_Column_Location = Next_Column_Location - Column_Spacing
Loop
half_spacing_behind = False
If Next_Column_Location + Column_Spacing - (Column_Spacing * 0.5) > include_location_behind The
num_of_columns_behind = num_of_columns_behind + 1
half_spacing_behind = True
```

```
End If
Next_Column_Location = 0 + Column_Spacing
Do Until Next_Column_Location > include_location_infront
num_of_columns_infront = num_of_columns_infront + 1
Next_Column_Location = Next_Column_Location + Column_Spacing
Loop
half_spacing_infront = False
If Next_Column_Location - Column_Spacing + (Column_Spacing * 0.5) < include_location_infront Then
num_of_columns_infront = num_of_columns_infront + 1
half_spacing_infront = True
End If
'-----------------------------------------------------------------------------
Select Case Inst_Type
Case "SINGLE"
Num_Of_Columns_Illuminance = num_of_columns_infront + num_of_columns_behind
Case "STAGGERED"
num_of_columns_infront = num_of_columns_infront * 2
If half_spacing_infront = True Then
num_of_columns_infront = num_of_columns_infront - 1
End If
num_of_columns_behind = num_of_columns_behind * 2
If half_spacing_behind = True Then
num_of_columns_behind = num_of_columns_behind - 1
End If
num_of_columns_behind = num_of_columns_behind - 1
Num_Of_Columns_Illuminance = num_of_columns_infront + num_of_columns_behind
Case "OPPOSITE"
num_of_columns_infront = num_of_columns_infront * 2
```

```
num_of_columns_behind = num_of_columns_behind * 2
Num_Of_Columns_Illuminance = num_of_columns_infront + num_of_columns_behind
Case "TWIN CENTRAL"
num_of_columns_infront = num_of_columns_infront * 2
num_of_columns_behind = num_of_columns_behind * 2
Num_Of_Columns_Illuminance = num_of_columns_infront + num_of_columns_behind
End Select
End Function
```

**APPENDIX G**

Procedures for calculating illuminance

---

```
'------------------writes point to column distances to sheet'------------------
'------------------x,y,z locations'------------------
For j = 1 To points_long
Worksheets("Luminaire" & m).Cells(17, j + 1).Value = longpoints(j)
Next j
For k = 1 To points_across
Worksheets("Luminaire" & m).Cells(22, k + 1).Value = widepoints(k)
Next k
For l = 1 To points_across
Worksheets("Luminaire" & m).Cells(27, l + 1).Value = widepoints(l)
Next l
'------------------x,y,z distances form column'------------------
ReDim longpoints_dist(points_long)
For j = 1 To points_long
longpoints_dist(j) = longpoints(j) - Worksheets("Luminaire" & m).Cells(6, 2).Value
Worksheets("Luminaire" & m).Cells(18, j + 1).Value = longpoints_dist(j)
Next j
ReDim widepoints_dist(points_across)
For k = 1 To points_across
widepoints_dist(k) = widepoints(k) - Worksheets("Luminaire" & m).Cells(7, 2).Value
Worksheets("Luminaire" & m).Cells(23, k + 1).Value = widepoints_dist(k)
Next k
```

```
For l = 1 To points_across
Worksheets("Luminaire" & m).Cells(28, l + 1).Value = column_height
Next l
'-------------------x',y',z' values'-------------------
With Worksheets("Luminaire" & m)
orientation = .Cells(10, 2).Value
ReDim longpoints_dist_primed(points_long)
For j = 1 To points_long
longpoints_dist_primed(j) = longpoints_dist(j) * Cos(Application.WorksheetFunction.Radians(orientation
.Cells(19, j + 1).Value = longpoints_dist_primed(j)
Next j
ReDim widepoints_dist_primed(points_across)
For k = 1 To points_across
widepoints_dist_primed(k) = widepoints_dist(k) * Cos(Application.WorksheetFunction.Radians(orientatio
Cos(Application.WorksheetFunction.Radians(tilt)) - column_height * Sin(Application.WorksheetFunction
.Cells(24, k + 1).Value = widepoints_dist_primed(k)
Next k
ReDim height_dist_primed(points_across)
For l = 1 To points_across
height_dist_primed(l) = 0 - .Cells(23, l + 1).Value * (0 - Cos(Application.WorksheetFunction.Radians(orie
Sin(Application.WorksheetFunction.Radians(tilt))) + column_height * Cos(Application.WorksheetFunctio
.Cells(29, l + 1).Value = height_dist_primed(l)
Next l
```

---

```
'----------------calculates and writes C-planes----------------
l = 37
```

```
k = 2
Do Until .Cells(36, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = MyAtan2(longpoints_dist_primed(k - 1), widepoints_dist_primed(l - 36))
l = l + 1
Loop
k = k + 1
l = 37
Loop
```

---

```
'----------------calculates and writes Gamma Planes----------------
l = 57
k = 2
Do Until .Cells(56, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = MyAtan2(height_dist_primed(l - 56), Sqr(longpoints_dist_primed(k - 1) ^ 2 + widepo
l = l + 1
Loop
k = k + 1
l = 57
Loop
```

---

```
'----------------calculates and writes Epsilon Planes----------------
l = 77
k = 2
```

```
Do Until .Cells(76, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = MyAtan2(column_height, Sqr(longpoints_dist(k - 1) ^ 2 + widepoints_dist(l - 76) ^ 2))
l = l + 1
Loop
k = k + 1
l = 77
Loop
```

---

```
'----------------calculates and writes intensities----------------
l = 97
k = 2
Do Until .Cells(96, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = Quadratic_Intensity(.Cells(l - 40, k).Value, .Cells(l - 60, k).Value)
l = l + 1
Loop
k = k + 1
l = 97
Loop
```

---

```
'----------------calculates and writes illuminances per 1000 lumens----------------
l = 117
k = 2
Do Until .Cells(116, k) = ""
```

```
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = .Cells(l - 20, k).Value * Cos(WorksheetFunction.Radians(.Cells(l - 40, k))) ^ 3 / colum
l = l + 1
Loop
k = k + 1
l = 117
Loop
```

---

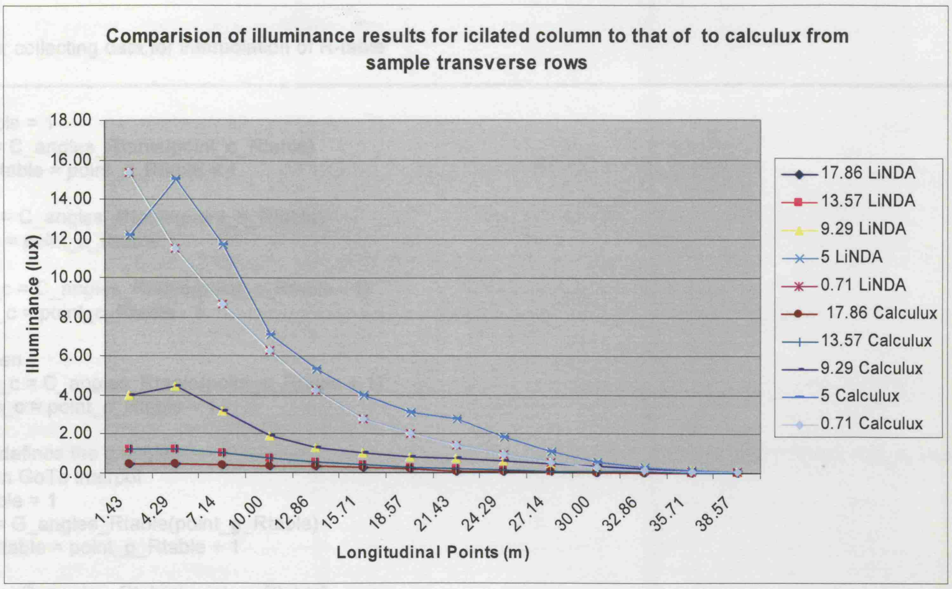
```
'----------------calculates and writes illuminances including lamp lumens----------------
l = 137
k = 2
Do Until .Cells(136, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = .Cells(l - 20, k).Value * (lamp_lumens * number_of_lamps) / 1000
l = l + 1
Loop
k = k + 1
l = 137
Loop
```

---

```
'----------------calculates and writes illuminances including lamp lumen and maintenance factor--------------
l = 157
k = 2
Do Until .Cells(156, k) = ""
Do Until .Cells(l, 1) = ""
```

```
.Cells(l, k).Value = .Cells(l - 20, k).Value * maintainance_factor
'.Cells(l, k).Value = 1 'check if result page is finding it
l = l + 1
Loop
k = k + 1
l = 157
Loop
```

## APPENDIX H


Comparision of illuminance results for icilated column to that of to calculux from sample transverse rows
Illuminance (lux)
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
1.43
4.29
7.14
10.00
12.86
15.71
18.57
21.43
24.29
27.14
30.00
32.86
35.71
38.57
Longitudinal Points (m)
17.86 LiNDA
13.57 LiNDA
9.29 LiNDA
5 LiNDA
0.71 LiNDA
17.86 Calculux
13.57 Calculux
9.29 Calculux
5 Calculux
0.71 Calculux

## APPENDIX I

Procedure for collecting data for interpolation of R-table

---

```
point_c_Rtable = 1
Do Until c <= C_angles_Rtable(point_c_Rtable)
   point_c_Rtable = point_c_Rtable + 1
Loop
mid_point_c = C_angles_Rtable(point_c_Rtable)
mid_value_c = point_c_Rtable
If c > 0 Then
lower_point_c = C_angles_Rtable(point_c_Rtable - 1)
lower_value_c = point_c_Rtable - 1
End If
If c < 165 Then
higher_point_c = C_angles_Rtable(point_c_Rtable + 1)
higher_value_c = point_c_Rtable + 1
End If
'----------------defines the g angles-------------------
If g > 12 Then GoTo interpol
point_g_Rtable = 1
Do Until g <= G_angles_Rtable(point_g_Rtable)
   point_g_Rtable = point_g_Rtable + 1
Loop
mid_point_g = G_angles_Rtable(point_g_Rtable)
```

```
mid_value_g = point_g_Rtable
If g > 0 Then
lower_point_g = G_angles_Rtable(point_g_Rtable - 1)
lower_value_g = point_g_Rtable - 1
End If
If g < 11.5 Then
higher_point_g = G_angles_Rtable(point_g_Rtable + 1)
higher_value_g = point_g_Rtable + 1
End If
interpol:
'--------------------------0 c 0 g---------------------------------------
If c = 0 And g = 0 Then
Quadratic_inten_Rtable = intensity_Rtable(1, 1)
Exit Function
End If
'--------------------------------normal c 0 g----------------------------------------------
If c <= 165 And c > 0 And g = 0 Then
low_c_interpolated = intensity_Rtable(lower_value_c, 1)
mid_c_interpolated = intensity_Rtable(mid_value_c, 1)
high_c_interpolated = intensity_Rtable(higher_value_c, 1)
Quadratic_inten_Rtable = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, low_c_interpola
high_c_interpolated)
Exit Function
End If
'----------------------------------high c 0 g----------------------------------------------------------
If c > 165 And c < 180 And g = 0 Then
low_c_interpolated = intensity_Rtable(lower_value_c, 1)
mid_c_interpolated = intensity_Rtable(mid_value_c, 1)
```

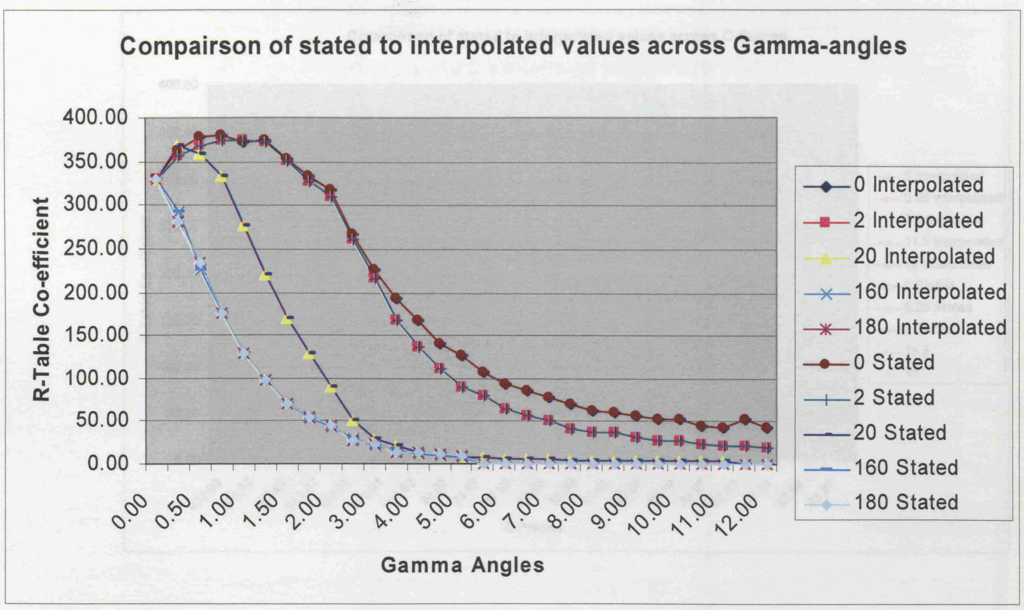
```
Quadratic_inten_Rtable = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_valu
intensity_Rtable(mid_value_c, 1))
Exit Function
End If
'--------------------------180c 0g-------------------------------------------------------
If c = 180 And g = 0 Then
Quadratic_inten_Rtable = intensity_Rtable(20, 1)
Exit Function
End If
'------------------------------0 c normal g-----------------------------
If c = 0 And g <= 11.5 And g > 0 Then
low_c_interpolated = intensity_Rtable(1, lower_value_g)
mid_c_interpolated = intensity_Rtable(1, mid_value_g)
high_c_interpolated = intensity_Rtable(1, higher_value_g)
Quadratic_inten_Rtable = Quadratic_I3(g, lower_point_g, mid_point_g, higher_point_g, low_c_interpolat
high_c_interpolated)
Exit Function
End If
'----------------normal c normal g---------------------------------------------
If c <= 165 And c > 0 And g <= 11.5 And g > 0 Then
low_c_interpolated = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, intensity_Rtable(lowe
intensity_Rtable(mid_value_c, lower_value_g), intensity_Rtable(higher_value_c, lower_value_g))
mid_c_interpolated = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, intensity_Rtable(lowe
intensity_Rtable(mid_value_c, mid_value_g), intensity_Rtable(higher_value_c, mid_value_g))
high_c_interpolated = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, intensity_Rtable(lowe
higher_value_g), intensity_Rtable(mid_value_c, higher_value_g), intensity_Rtable(higher_value_c, highe
Quadratic_inten_Rtable = Quadratic_I3(g, lower_point_g, mid_point_g, higher_point_g, low_c_interpolat
high_c_interpolated)
```

```
Exit Function
End If
'-----------------------------high c normal g---------------------------------------------
If c > 165 And g <= 11.5 And g > 0 Then
low_c_interpolated = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_value_
intensity_Rtable(mid_value_c, lower_value_g))
mid_c_interpolated = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_value_
intensity_Rtable(mid_value_c, mid_value_g))
high_c_interpolated = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_value_
intensity_Rtable(mid_value_c, higher_value_g))
Quadratic_inten_Rtable = Quadratic_I3(g, lower_point_g, mid_point_g, higher_point_g, low_c_interpola
high_c_interpolated)
Exit Function
End If
'-------------------------------180 c normal g----------------------------------------
If c = 180 And g > 0 And g <= 11.5 Then
low_c_interpolated = intensity_Rtable(20, lower_value_g)
mid_c_interpolated = intensity_Rtable(20, mid_value_g)
high_c_interpolated = intensity_Rtable(20, higher_value_g)
Quadratic_inten_Rtable = Quadratic_I3(g, lower_point_g, mid_point_g, higher_point_g, low_c_interpola
high_c_interpolated)
Exit Function
End If
'------------------------------0 c high g-----------------------------
If c = 0 And g > 11.5 And g < 12 Then
low_c_interpolated = intensity_Rtable(1, lower_value_g)
mid_c_interpolated = intensity_Rtable(1, mid_value_g)
Quadratic_inten_Rtable = linear_interpolation(g, lower_point_g, mid_point_g, low_c_interpolated, mid_
```

```
Exit Function
End If
'----------------------------normal c high g------------------------------------------
If c <= 165 And c > 0 And g > 11.5 And g < 12 Then
low_c_interpolated = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, intensity_Rtable(lowe
intensity_Rtable(mid_value_c, lower_value_g), intensity_Rtable(higher_value_c, lower_value_g))
mid_c_interpolated = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, intensity_Rtable(lowe
intensity_Rtable(mid_value_c, mid_value_g), intensity_Rtable(higher_value_c, mid_value_g))
Quadratic_inten_Rtable = linear_interpolation(g, lower_point_g, mid_point_g, low_c_interpolated, mid_c
Exit Function
End If
'------------------------------high c high g-------------------------------------------------------------------
If c > 165 And g < 180 And g > 11.5 And g < 12 Then
low_c_interpolated = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_value_c
intensity_Rtable(mid_value_c, lower_value_g))
mid_c_interpolated = linear_interpolation(c, lower_point_c, mid_point_c, intensity_Rtable(lower_value_c
intensity_Rtable(mid_value_c, mid_value_g))
Quadratic_inten_Rtable = linear_interpolation(g, lower_point_g, mid_point_g, low_c_interpolated, mid_c
Exit Function
End If
'---------------------------180 c and high g--------------------------------------------------
If c = 180 And g > 11.5 And g < 12 Then
low_c_interpolated = intensity_Rtable(20, lower_value_g)
mid_c_interpolated = intensity_Rtable(20, mid_value_g)
Quadratic_inten_Rtable = linear_interpolation(g, lower_point_g, mid_point_g, low_c_interpolated, mid_c
Exit Function
End If
'----------------------------------0 c 12 g-------------------------------------------
```

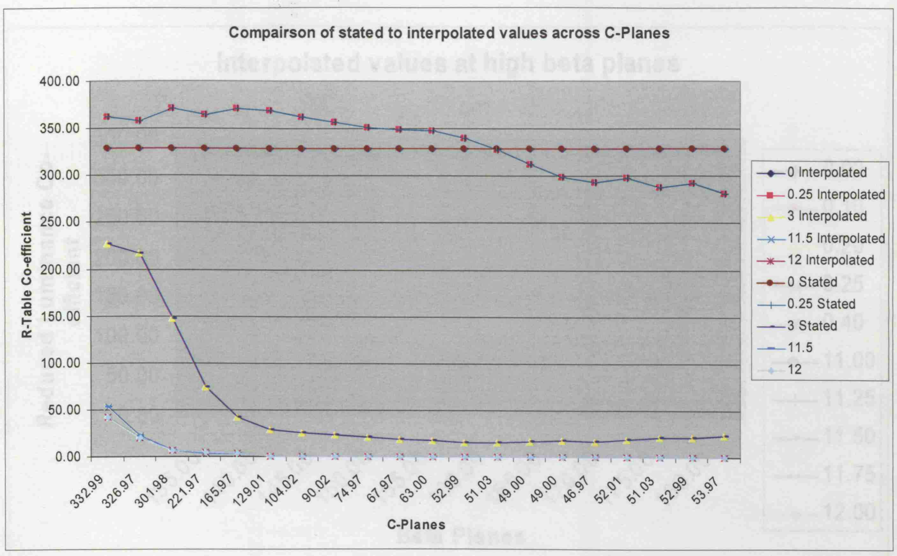
```
If c = 0 And g = 12 Then
Quadratic_inten_Rtable = intensity_Rtable(1, 29)
Exit Function
End If
'-------------------------------------normal c and 12 g---------------------------------
If c > 0 And c <= 165 And g = 12 Then
low_c_interpolated = intensity_Rtable(lower_value_c, 29)
mid_c_interpolated = intensity_Rtable(mid_value_c, 29)
high_c_interpolated = intensity_Rtable(higher_value_c, 29)
Quadratic_inten_Rtable = Quadratic_I3(c, lower_point_c, mid_point_c, higher_point_c, low_c_interpola
high_c_interpolated)
Exit Function
End If
If c > 165 And c < 180 And g = 12 Then
low_c_interpolated = intensity_Rtable(lower_value_c, 29)
mid_c_interpolated = intensity_Rtable(mid_value_c, 29)
Quadratic_inten_Rtable = linear_interpolation(c, lower_point_c, mid_point_c, low_c_interpolated, mid_
Exit Function
End If
'-------------------------180 c 12 g--------------------------------------
If c = 180 And g = 12 Then
Quadratic_inten_Rtable = intensity_Rtable(20, 29)
Exit Function
End If
'-----------------------------------g>12-------------------------------------------------
If g > 12 Then
Quadratic_inten_Rtable = 0
End If
```

APPENDIX J.1



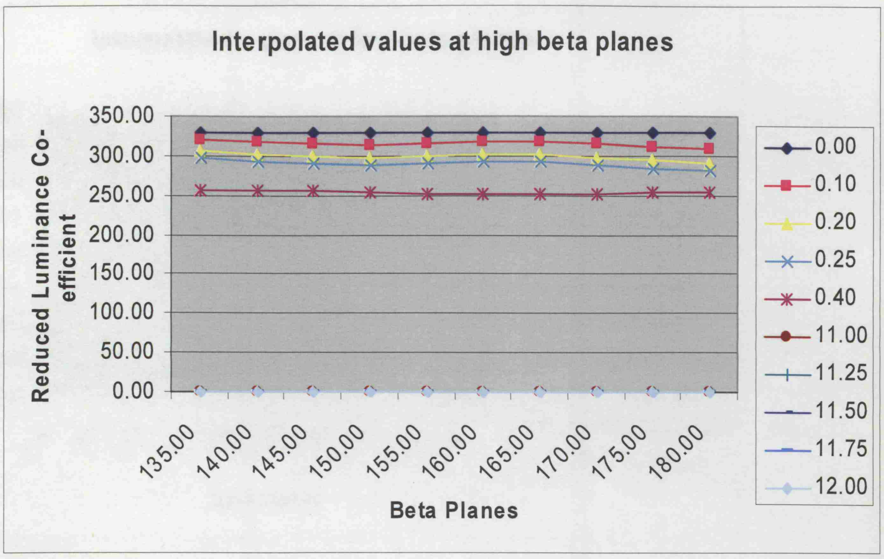
Compairson of stated to interpolated values across Gamma-angles
R-Table Co-efficient
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
0.00
0.50
1.00
1.50
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
Gamma Angles
0 Interpolated
2 Interpolated
20 Interpolated
160 Interpolated
180 Interpolated
0 Stated
2 Stated
20 Stated
160 Stated
180 Stated

## APPENDIX J.2



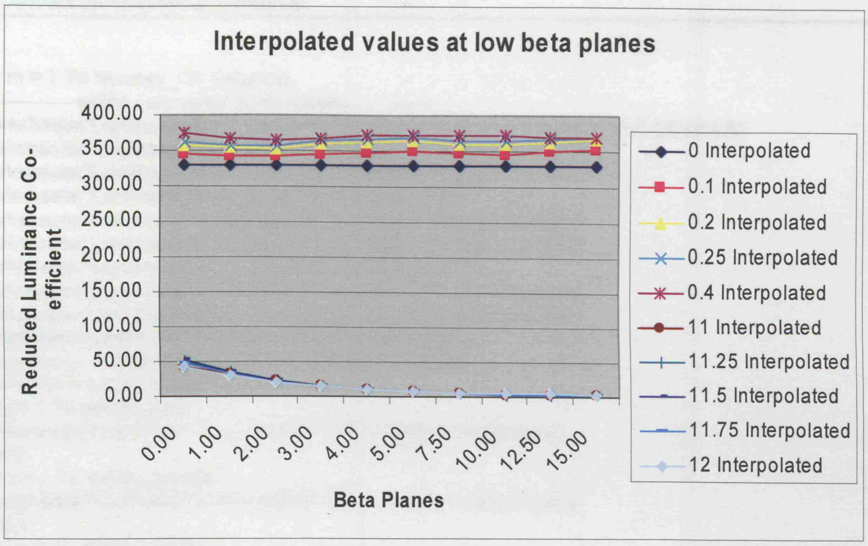
Comparison of stated to interpolated values across C-Planes
R-Table Co-efficient
C-Planes
0 Interpolated
0.25 Interpolated
3 Interpolated
11.5 Interpolated
12 Interpolated
0 Stated
0.25 Stated
3 Stated
11.5
12

APPENDIX J.3



Interpolated values at high beta planes
Reduced Luminance Co-efficient
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
135.00
140.00
145.00
150.00
155.00
160.00
165.00
170.00
175.00
180.00
Beta Planes
0.00
0.10
0.20
0.25
0.40
11.00
11.25
11.50
11.75
12.00

## APPENDIX J.4



Interpolated values at low beta planes
Reduced Luminance Co-efficient
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
7.50
10.00
12.50
15.00
Beta Planes
0 Interpolated
0.1 Interpolated
0.2 Interpolated
0.25 Interpolated
0.4 Interpolated
11 Interpolated
11.25 Interpolated
11.5 Interpolated
11.75 Interpolated
12 Interpolated

## APPENDIX K

Procedures for calculating luminance

```
For m = 1 To Number_Of_Columns
'---------------writes calculation point details-------------
Worksheets("Luminaire" & m).Cells(15, 1).Value = "CALCULATION POINT DETAILS"
Worksheets("Luminaire" & m).Cells(17, 1).Value = "X point location"
Worksheets("Luminaire" & m).Cells(18, 1).Value = "X dist to lantern"
Worksheets("Luminaire" & m).Cells(19, 1).Value = "X'"
Worksheets("Luminaire" & m).Cells(22, 1).Value = "Y point location"
Worksheets("Luminaire" & m).Cells(23, 1).Value = "Y dist to lantern"
Worksheets("Luminaire" & m).Cells(24, 1).Value = "Y'"
Worksheets("Luminaire" & m).Cells(27, 1).Value = "Y point location"
Worksheets("Luminaire" & m).Cells(28, 1).Value = "Z dist to lantern"
Worksheets("Luminaire" & m).Cells(29, 1).Value = "Z'"
'-------------------writes point to column distances to sheet'-------------------
'-------------------x,y,z locations'-------------------
For j = 1 To points_long
Worksheets("Luminaire" & m).Cells(17, j + 1).Value = longpoints(j)
Next j
For k = 1 To points_across
Worksheets("Luminaire" & m).Cells(22, k + 1).Value = widepoints(k)
Next k
For l = 1 To points_across
Worksheets("Luminaire" & m).Cells(27, l + 1).Value = widepoints(l)
```

```
Next l
'------------------x,y,z distances form column'------------------
ReDim longpoints_dist(points_long)
For j = 1 To points_long
longpoints_dist(j) = longpoints(j) - Worksheets("Luminaire" & m).Cells(6, 2).Value
Worksheets("Luminaire" & m).Cells(18, j + 1).Value = longpoints_dist(j)
Next j
ReDim widepoints_dist(points_across)
For k = 1 To points_across
widepoints_dist(k) = widepoints(k) - Worksheets("Luminaire" & m).Cells(7, 2).Value
Worksheets("Luminaire" & m).Cells(23, k + 1).Value = widepoints_dist(k)
Next k
For l = 1 To points_across
Worksheets("Luminaire" & m).Cells(28, l + 1).Value = column_height
Next l
'------------------x',y',z' values'------------------
With Worksheets("Luminaire" & m)
orientation = .Cells(10, 2).Value
ReDim longpoints_dist_primed(points_long)
For j = 1 To points_long
longpoints_dist_primed(j) = longpoints_dist(j) * Cos(Application.WorksheetFunction.Radians(orientation))
.Cells(19, j + 1).Value = longpoints_dist_primed(j)
Next j
ReDim widepoints_dist_primed(points_across)
For k = 1 To points_across
widepoints_dist_primed(k) = widepoints_dist(k) * Cos(Application.WorksheetFunction.Radians(orientatio
Cos(Application.WorksheetFunction.Radians(tilt)) - column_height * Sin(Application.WorksheetFunction.
.Cells(24, k + 1).Value = widepoints_dist_primed(k)
```

```
Next k
ReDim height_dist_primed(points_across)
For l = 1 To points_across
height_dist_primed(l) = 0 - .Cells(23, l + 1).Value * (0 - Cos(Application.WorksheetFunction.Radians(ori
Sin(Application.WorksheetFunction.Radians(tilt))) + column_height * Cos(Application.WorksheetFuncti
.Cells(29, l + 1).Value = height_dist_primed(l)
Next l
```

---

```
'C Planes'-----------------------------------------------
l = 37
k = 2
Do Until .Cells(36, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = MyAtan2(longpoints_dist_primed(k - 1), widepoints_dist_primed(l - 36))
l = l + 1
Loop
k = k + 1
l = 37
Loop
```

---

```
'Gamma Planes---------------------------------
l = 57
k = 2
Do Until .Cells(56, k) = ""
Do Until .Cells(l, 1) = ""
```

```
.Cells(l, k).Value = MyAtan2(height_dist_primed(l - 56), Sqr(longpoints_dist_primed(k - 1) ^ 2 + widepoin
l = l + 1
Loop
k = k + 1
l = 57
Loop
```

---

```
'Epsilon Planes---------------------------------
l = 77
k = 2
Do Until .Cells(76, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = MyAtan2(column_height, Sqr(longpoints_dist(k - 1) ^ 2 + widepoints_dist(l - 76) ^ 2))
l = l + 1
Loop
k = k + 1
l = 77
Loop
```

---

```
'----------------Tan Epsilon  Planes----------------
l = 97
k = 2
Do Until .Cells(96, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = Tan(Application.WorksheetFunction.Radians(.Cells(l - 20, k).Value))
```

```
l = l + 1
Loop
k = k + 1
l = 97
Loop
```

---

```
'Intensities---------------------------------
l = 117
k = 2
Do Until .Cells(116, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = Quadratic_Intensity(.Cells(l - 60, k).Value, .Cells(l - 80, k).Value)
l = l + 1
Loop
k = k + 1
l = 117
Loop
```

---

```
'----------------calculates and writes Beta----------------
l = 137 + sheet_loc
k = 2
Do Until .Cells(136 + sheet_loc, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = beta(-60, Obs_Trans_Loc(p), longpoints(k - 1), widepoints(l - 136 - sheet_loc), .Cell
2).Value)
```

```
l = l + 1
Loop
k = k + 1
l = 137 + sheet_loc
Loop
```

---

```
'RTABLE--------------------------------
'----------------calculates and writes values----------------
l = 157 + sheet_loc
k = 2
Do Until .Cells(156 + sheet_loc, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = Inten_Rtable(.Cells(l - 20, k), .Cells(l - sheet_loc - 60, k))
l = l + 1
Loop
k = k + 1
l = 157 + sheet_loc
Loop
```

---

```
'----------------calculates and writes luminance per 1000 lumens----------------
l = 177 + sheet_loc
k = 2
Do Until .Cells(176 + sheet_loc, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = (.Cells(l - 60 - sheet_loc, k) * .Cells(l - 20, k) * 0.0001) / column_height ^ 2
```

```
l = l + 1
Loop
k = k + 1
l = 177 + sheet_loc
Loop
```

---

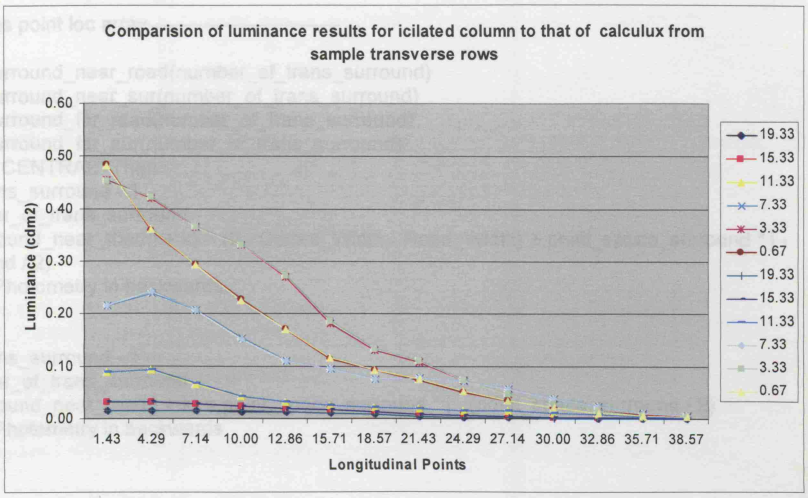
```
'----------------luminance including lamp lumens----------------
l = 197 + sheet_loc
k = 2
Do Until .Cells(196 + sheet_loc, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = .Cells(l - 20, k).Value * (lamp_lumens * number_of_lamps) / 1000
'.Cells(l, k).Value = -1 ' can use to check results details is finding value
l = l + 1
Loop
k = k + 1
l = 197 + sheet_loc
Loop
```

---

```
' ---------------------------- luminance including lamp lumens and MF---------------------------------
l = 217 + sheet_loc
k = 2
Do Until .Cells(216 + sheet_loc, k) = ""
Do Until .Cells(l, 1) = ""
.Cells(l, k).Value = .Cells(l - 20, k).Value * maintainance_factor
```

```
'.Cells(l, k).Value = 1 ' can use to check results details is finding value
l = l + 1
Loop
k = k + 1
l = 217 + sheet_loc
Loop
Next p
End With
Next m
```

## APPENDIX L



Comparision of luminance results for icilated column to that of calculux from sample transverse rows
Luminance (cdm2)
0.60
0.50
0.40
0.30
0.20
0.10
0.00
1.43
4.29
7.14
10.00
12.86
15.71
18.57
21.43
24.29
27.14
30.00
32.86
35.71
38.57
Longitudinal Points
19.33
15.33
11.33
7.33
3.33
0.67
19.33
15.33
11.33
7.33
3.33
0.67

## APPENDIX M

Section showing calculation points for surround ratio being defined

---

```
 '---------------------writes point loc array----------------------------
Dim p, m As Integer
ReDim widepoints_surround_near_road(number_of_trans_surround)
ReDim widepoints_surround_near_sur(number_of_trans_surround)
ReDim widepoints_surround_far_road(number_of_trans_surround)
ReDim widepoints_surround_far_sur(number_of_trans_surround)
If Inst_Type = "TWIN CENTRAL" Then
   k = number_of_trans_surround - 1
   For j = 1 To number_of_trans_surround
      widepoints_surround_near_road(j + k) = (0 - Centre_Width - Road_Width) + point_space_surround
(point_space_surround / 2)
      k = k - 2 'reads Photometry in backwards
   Next j
Else
   k = number_of_trans_surround - 1
   For j = 1 To number_of_trans_surround
      widepoints_surround_near_road(j + k) = point_space_surround * j - (point_space_surround / 2)
      k = k - 2 'reads Photometry in backwards
   Next j
End If
'------------------
If Inst_Type = "TWIN CENTRAL" Then
```

```
   For j = 1 To number_of_trans_surround
      widepoints_surround_near_sur(j) = 0 - Road_Width - Centre_Width - (point_space_surround * j) +
   Next j
Else
   For j = 1 To number_of_trans_surround
      widepoints_surround_near_sur(j) = 0 - point_space_surround * j + (point_space_surround / 2)
   Next j
End If
k = number_of_trans_surround - 1
For j = 1 To number_of_trans_surround
   widepoints_surround_far_sur(j + k) = Road_Width + point_space_surround * j - (point_space_surrou
   k = k - 2 'reads Photometry in backwards
Next j
For j = 1 To number_of_trans_surround
   widepoints_surround_far_road(j) = Road_Width - point_space_surround * j + (point_space_surround
Next j
'----------------------------------------------------------
ReDim road_surround_points(number_of_trans_surround * 2)
ReDim surr_surround_points(number_of_trans_surround * 2)
For i = 1 To number_of_trans_surround
   road_surround_points(i) = widepoints_surround_far_road(i)
Next i
Do Until i > number_of_trans_surround * 2
   road_surround_points(i) = widepoints_surround_near_road(i - number_of_trans_surround)
i = i + 1
Loop
For i = 1 To number_of_trans_surround
   surr_surround_points(i) = widepoints_surround_far_sur(i)
```

```
Next i
Do Until i > number_of_trans_surround * 2
   surr_surround_points(i) = widepoints_surround_near_sur(i - number_of_trans_surround)
   i = i + 1
Loop
```

## APPENDIX N

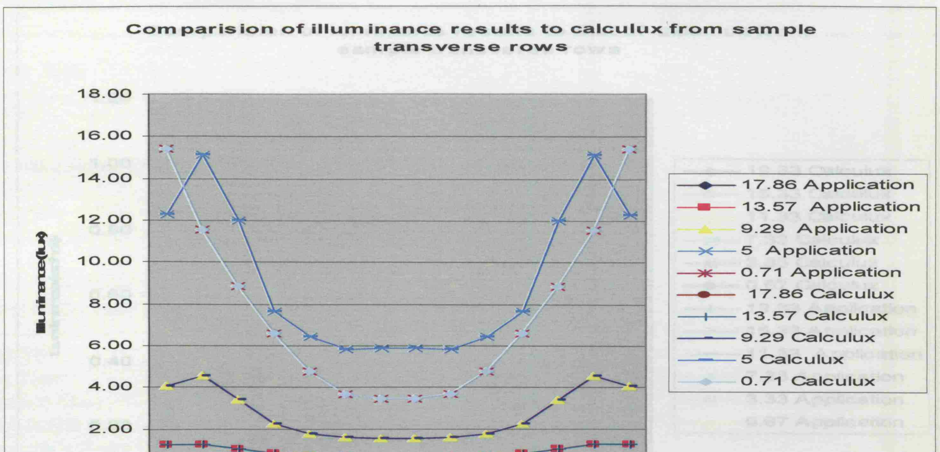
Section showing quality figures for total results being generated

---

```
'luminance Results---------------------------------
'----------------writes results----------------
l = 67 + sheet_loc
k = 2
Do Until .Cells(66 + sheet_loc, k + 1) = ""
Do Until .Cells(l + 1, 1) = ""
illuminance = 0
For m = 1 To Number_Of_Columns
illuminance = illuminance + Worksheets("luminaire" & m).Cells(l - sheet_loc + 150 + other_sheet_loc, k)
Next m
.Cells(l + 1, k + 1).Value = illuminance
Worksheets("data").Cells(l - 66 - sheet_loc + 10 + (points_across * (p - 1)), k - 1).Value = illuminance
l = l + 1
Loop
k = k + 1
l = 67 + sheet_loc
Loop
'------------------------QUALITY FIGURES----------------------------------
.Cells(84 + sheet_loc, 1).Value = "AVERAGE LUMINANCE"
.Cells(84 + sheet_loc, 5).Value = "MINIMUM LUMINANCE"
.Cells(84 + sheet_loc, 9).Value = "MAXIMUM LUMINANCE"
.Cells(84 + sheet_loc, 13).Value = "MINIMUM/AVERAGE"
```

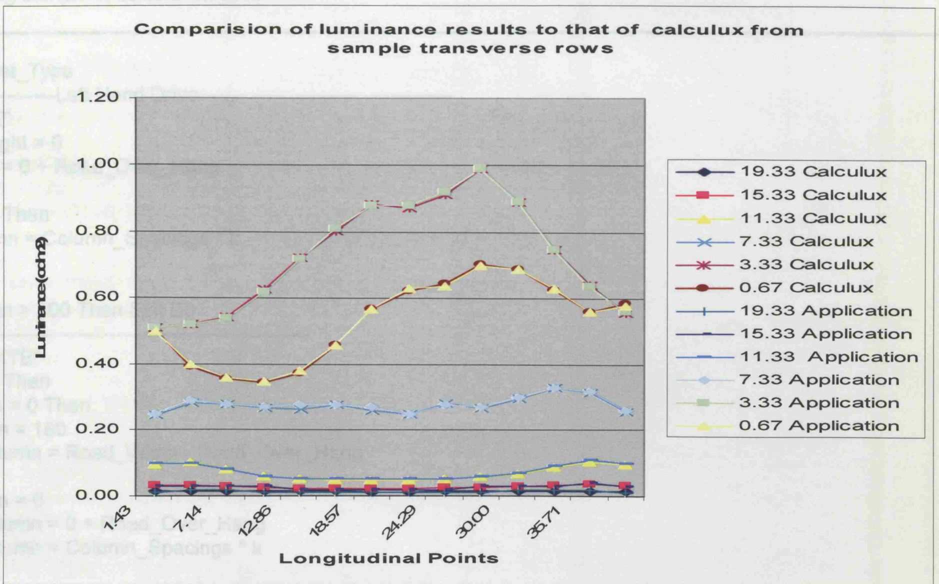
```
.Select
.Cells(68 + sheet_loc, 3).Select
Selection.CurrentRegion.Select 'selecting area
.Cells(84 + sheet_loc, 2).Value = Application.WorksheetFunction.Average(Selection)
.Cells(84 + sheet_loc, 6).Value = Application.WorksheetFunction.Min(Selection)
.Cells(84 + sheet_loc, 10).Value = Application.WorksheetFunction.Max(Selection)
.Cells(84 + sheet_loc, 14).Value = .Cells(84 + sheet_loc, 6).Value / .Cells(84 + sheet_loc, 2).Value
If .Cells(84 + sheet_loc, 2).Value < Ave_Lum Then
Ave_Lum = .Cells(84 + sheet_loc, 2).Value
End If
If .Cells(84 + sheet_loc, 14).Value < Uo_Lum Then
Uo_Lum = .Cells(84 + sheet_loc, 14).Value
End If
'----------------------Ul-------------------------------------------
j = 1
For i = 1 To No_Of_Obs
Do Until .Cells(j + 67 + sheet_loc, 1).Value = Obs_Trans_Loc(i)
j = j + 1
Loop
Range(.Cells(j + 67 + sheet_loc, 3), .Cells(j + 67 + sheet_loc, 2 + points_long)).Select
low_Ul = Application.WorksheetFunction.Min(Selection)
high_Ul = Application.WorksheetFunction.Max(Selection)
.Cells(j + 67 + sheet_loc, 4 + points_long) = "Ul"
.Cells(j + 67 + sheet_loc, 5 + points_long) = low_Ul / high_Ul
If Ul > .Cells(j + 67 + sheet_loc, 5 + points_long).Value Then Ul = .Cells(j + 67 + sheet_loc, 5 + points_lo
Next i
Next p
```

## APPENDIX O.1



Comparision of illuminance results to calculux from sample transverse rows
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
Illuminance(lux)
1.43
7.14
12.86
18.57
24.29
30.00
35.71
Longitudinal Points (m)
17.86 Application
13.57 Application
9.29 Application
5 Application
0.71 Application
17.86 Calculux
13.57 Calculux
9.29 Calculux
5 Calculux
0.71 Calculux

## APPENDIX 0.2



Comparision of luminance results to that of calculux from sample transverse rows
Luminance(cd/m2)
1.20
1.00
0.80
0.60
0.40
0.20
0.00
1.43
7.14
12.86
18.57
24.29
30.00
35.71
Longitudinal Points
19.33 Calculux
15.33 Calculux
11.33 Calculux
7.33 Calculux
3.33 Calculux
0.67 Calculux
19.33 Application
15.33 Application
11.33 Application
7.33 Application
3.33 Application
0.67 Application

**APPENDIX P**

Section showing sample of calculation of Lv

---

```
Select Case Inst_Type
'--------------------------Left Hand Drive------------------------------------
Case "SINGLE"
Lv_percent_Right = 0
Y_loc_column = 0 + Road_Over_Hang
orientation = 0
If start = False Then
   X_loc_column = Column_Spacings * k
   k = k + 1
End If
If X_loc_column > 500 Then Exit Do
'------------------------------------------------------------
Case "OPPOSITE"
If start = False Then
   If orientation = 0 Then
      orientation = 180
      Y_loc_column = Road_Width - Road_Over_Hang
   Else
      orientation = 0
      Y_loc_column = 0 + Road_Over_Hang
      X_loc_column = Column_Spacings * k
      k = k + 1
```

```
   End If
Else
   Y_loc_column = 0 + Road_Over_Hang
orientation = 0
End If
If X_loc_column > 500 Then Exit Do
'---------------------------------------------------------------
Case "TWIN CENTRAL"
If start = False Then
   If orientation = 0 Then
      orientation = 180
      Y_loc_column = Road_Width - Road_Over_Hang
   Else
      orientation = 0
      Y_loc_column = Road_Width + Road_Over_Hang + Centre_Width
      X_loc_column = Column_Spacings * k
      k = k + 1
   End If
Else
   Y_loc_column = Road_Width + Road_Over_Hang + Centre_Width
   orientation = 0
End If
If X_loc_column > 500 Then Exit Do
'---------------------------------------------------------------
Case "STAGGERED"
If start = False Then
   If orientation = 0 Then
      orientation = 180
```

```
      Y_loc_column = Road_Width - Road_Over_Hang
      X_loc_column = staggered_spacing * k
      k = k + 1
   Else
      orientation = 0
      Y_loc_column = 0 + Road_Over_Hang
      X_loc_column = staggered_spacing * k
      k = k + 1
   End If
Else
   Y_loc_column = 0 + Road_Over_Hang
   orientation = 0
End If
If X_loc_column > 500 Then Exit Do
End Select
Else
Select Case Inst_Type
'--------------------Right Hand Drive----------------------------------------
Case "SINGLE"
Lv_percent_Left = 0
Y_loc_column = Road_Width - Road_Over_Hang
orientation = 180
If start = False Then
   X_loc_column = Column_Spacings * k
   k = k + 1
End If
If X_loc_column > 500 Then Exit Do
'--------------------------------------------------------------
```

```
Case "OPPOSITE"
If start = False Then
   If orientation = 0 Then
      orientation = 180
      Y_loc_column = Road_Width - Road_Over_Hang
   Else
      orientation = 0
      Y_loc_column = 0 + Road_Over_Hang
      X_loc_column = Column_Spacings * k
      k = k + 1
   End If
Else
   Y_loc_column = 0 + Road_Over_Hang
orientation = 0
End If
If X_loc_column > 500 Then Exit Do
'------------------------------------------------------------------
Case "TWIN CENTRAL"
If start = False Then
   If orientation = 0 Then
      orientation = 180
      Y_loc_column = Road_Width - Road_Over_Hang
   Else
      orientation = 0
      Y_loc_column = Road_Width + Road_Over_Hang + Centre_Width
      X_loc_column = Column_Spacings * k
      k = k + 1
   End If
```

```
Else
   Y_loc_column = Road_Width + Road_Over_Hang + Centre_Width
   orientation = 0
End If
If X_loc_column > 500 Then Exit Do
'------------------------------------------------------------
Case "STAGGERED"
If start = False Then
   If orientation = 180 Then
      orientation = 0
      Y_loc_column = 0 + Road_Over_Hang
      X_loc_column = staggered_spacing * k
      k = k + 1
   Else
      orientation = 180
      Y_loc_column = Road_Width - Road_Over_Hang
      X_loc_column = staggered_spacing * k
      k = k + 1
   End If
Else
   Y_loc_column = Road_Width - Road_Over_Hang
   orientation = 180
End If
If X_loc_column > 500 Then Exit Do
End Select
End If
'-----------------locations-----------------------------
'.Cells(90 + sheet_loc + loc_correct + 80, 1 + i).Value = "Col " & i
```

```
.Cells(87 + sheet_loc + loc_correct + 80, 2).Value = X_loc_obs
.Cells(88 + sheet_loc + loc_correct + 80, 2).Value = Obs_Trans_Loc(j)
.Cells(89 + sheet_loc + loc_correct + 80, 2).Value = 1.5
.Cells(91 + sheet_loc + loc_correct + 80, 1 + i).Value = X_loc_column
.Cells(92 + sheet_loc + loc_correct + 80, 1 + i).Value = Y_loc_column
.Cells(93 + sheet_loc + loc_correct + 80, 1 + i).Value = column_height
.Cells(94 + sheet_loc + loc_correct + 80, 1 + i).Value = orientation
'-----------------distances------------------------------
x_distance = X_loc_obs - X_loc_column
.Cells(96 + sheet_loc + loc_correct + 80, 1 + i).Value = x_distance
y_distance = Obs_Trans_Loc(j) - Y_loc_column
.Cells(97 + sheet_loc + loc_correct + 80, 1 + i).Value = y_distance
z_distance = column_height - 1.5
.Cells(98 + sheet_loc + loc_correct + 80, 1 + i).Value = z_distance
'-----------------distances primed------------------------------
.Cells(100 + sheet_loc + loc_correct + 80, 1 + i).Value = .Cells(96 + sheet_loc + loc_correct + 80, 1 + i)
Cos(Application.WorksheetFunction.Radians(orientation))
.Cells(101 + sheet_loc + loc_correct + 80, 1 + i).Value = .Cells(97 + sheet_loc + loc_correct + 80, 1 + i)
Cos(Application.WorksheetFunction.Radians(orientation)) * Cos(Application.WorksheetFunction.Radian
Sin(Application.WorksheetFunction.Radians(tilt))
.Cells(102 + sheet_loc + loc_correct + 80, 1 + i).Value = 0 - .Cells(97 + sheet_loc + loc_correct + 80, 1
Cos(Application.WorksheetFunction.Radians(orientation)) * Sin(Application.WorksheetFunction.Radian
sheet_loc + loc_correct + 80, 1 + i).Value * Cos(Application.WorksheetFunction.Radians(tilt))
'-------------------------cplanes etc------------------------------
.Cells(107 + sheet_loc + loc_correct + 80, 1 + i).Value = MyAtan2(.Cells(100 + sheet_loc + loc_correct
.Cells(101 + sheet_loc + loc_correct + 80, 1 + i).Value)
.Cells(108 + sheet_loc + loc_correct + 80, 1 + i).Value = MyAtan2(.Cells(102 + sheet_loc + loc_correct
Sqr(.Cells(100 + sheet_loc + loc_correct + 80, 1 + i).Value ^ 2 + .Cells(101 + sheet_loc + loc_correct +
```

```
intensity = Quadratic_Intensity(.Cells(108 + sheet_loc + loc_correct + 80, 1 + i).Value, .Cells(107 + shee
+ i).Value) * (lamp_lumens / 1000)
.Cells(109 + sheet_loc + loc_correct + 80, 1 + i).Value = intensity
'-----------------lux and angle----------------------------------
eye_dist = Sqr(Sqr(x_distance ^ 2 + y_distance ^ 2) ^ 2 + z_distance ^ 2)
.Cells(111 + sheet_loc + loc_correct + 80, 1 + i).Value = eye_dist
E_angle = A_Eye(X_loc_obs, Obs_Trans_Loc(j), 1.5, X_loc_column, Y_loc_column, column_height)
.Cells(112 + sheet_loc + loc_correct + 80, 1 + i).Value = E_angle
temp_cos = Cos(Application.WorksheetFunction.Radians(E_angle))
temp_lux = intensity * temp_cos / eye_dist ^ 2
.Cells(113 + sheet_loc + loc_correct + 80, 1 + i).Value = temp_lux
'-----------------------------------------------
temp_Lv = 10 * temp_lux / E_angle ^ 2
.Cells(115 + sheet_loc + loc_correct + 80, i + 1).Value = temp_Lv
If start = True Then
   .Cells(116 + sheet_loc + loc_correct + 80, 2).Value = 100
   .Cells(117 + sheet_loc + loc_correct + 80, 2).Value = temp_Lv
Else
   If orientation = 0 Then
      Lv_percent_Left = (temp_Lv / .Cells(117 + sheet_loc + loc_correct + 80, i).Value) * 100
      .Cells(116 + sheet_loc + loc_correct + 80, i + 1).Value = Lv_percent_Left
   Else
      Lv_percent_Right = (temp_Lv / .Cells(117 + sheet_loc + loc_correct + 80, i).Value) * 100
      .Cells(116 + sheet_loc + loc_correct + 80, i + 1).Value = Lv_percent_Right
   End If
End If
If orientation = 0 Then
   If Lv_percent_Left < 2 Then
```

```
        If Lv_percent_Right < 2 Then
            .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value = .Cells(117 + sheet_loc + loc_correct + 8
            .Cells(118 + sheet_loc + loc_correct + 80, i + 1).Value = "Rejected"
            Exit Do
        Else
            .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value = .Cells(117 + sheet_loc + loc_correct + 8
            .Cells(118 + sheet_loc + loc_correct + 80, i + 1).Value = "Rejected"
            GoTo next_loop
        End If
    End If
Else
    If Lv_percent_Right < 2 Then
        If Lv_percent_Left < 2 Then
            .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value = .Cells(117 + sheet_loc + loc_correct + 8
            .Cells(118 + sheet_loc + loc_correct + 80, i + 1).Value = "Rejected"
            Exit Do
        Else
            .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value = .Cells(117 + sheet_loc + loc_correct + 8
            .Cells(118 + sheet_loc + loc_correct + 80, i + 1).Value = "Rejected"
            GoTo next_loop
        End If
    End If
End If
If start = True Then
    .Cells(117 + sheet_loc + loc_correct + 80, 2).Value = temp_Lv
Else
    .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value = .Cells(117 + sheet_loc + loc_correct + 80, i).
End If
```

```
If Lv_Total < .Cells(117 + sheet_loc + loc_correct + 80, i + 1).Value Then Lv_Total = .Cells(117 + sheet_
1).Value
next_loop:
i = i + 1
start = False
Loop
Next j
.Select
.Cells(1, 1).Select
End With
results_summary_Lum
End Sub
```

## APPENDIX R

Code used to generate *Results Summary*

---

```
Sub results_summary_Lum()
With Worksheets("Results Summary")
'--------------clear the worksheet---------------------
.Cells.ClearContents
'-----------scheme details-------------------------
.Cells(1, 1).Value = "Lighting 'N' Design Application (LiNDA)"
.Cells(5, 1).Value = "CALCULATION OPTIONS"
.Cells(15, 1).Value = "SCHEME DETAILS"
.Cells(17, 1).Value = "INTERPOLATION OF PHOTOMETRY"
If Quad_Inten = True Then
.Cells(17, 2).Value = "QUADRATIC"
Else
.Cells(17, 2).Value = "LINEAR"
End If
.Cells(18, 1).Value = "CALCULATION TYPE"
.Cells(18, 2).Value = "LUMINANCE"
.Cells(20, 1).Value = "DRIVING SIDE"
If Left_Drive = True Then
.Cells(20, 2).Value = "LEFT"
Else
.Cells(20, 2).Value = "RIGHT"
End If
```

```
.Cells(21, 1).Value = "INSTALATION"
.Cells(21, 2).Value = Inst_Type
.Cells(22, 1).Value = "NUMBER OF LANES"
.Cells(22, 2).Value = Number_of_Lanes
.Cells(23, 1).Value = "ROAD SURFACE"
.Cells(23, 2).Value = Worksheets("Rtable").Cells(1, 1).Value
.Cells(24, 1).Value = "Q0 of Table"
.Cells(24, 2).Value = Worksheets("Rtable").Cells(7, 4).Value
.Cells(26, 1).Value = "ROAD WIDTH"
.Cells(26, 2).Value = Road_Width
.Cells(26, 3).Value = "metres"
.Cells(27, 1).Value = "COLUMN SPACINGS"
.Cells(27, 2).Value = Column_Spacings
.Cells(27, 3).Value = "metres"
.Cells(28, 1).Value = "COLUMN HEIGHT"
.Cells(28, 2).Value = column_height
.Cells(28, 3).Value = "metres"
.Cells(29, 1).Value = "LANTERN TILT"
.Cells(29, 2).Value = tilt
.Cells(29, 3).Value = "degrees"
.Cells(30, 1).Value = "OVERHANG"
.Cells(30, 2).Value = Road_Over_Hang
.Cells(30, 3).Value = "metres"
If Inst_Type = "TWIN CENTRAL" Then
.Cells(31, 1).Value = "CENTRAL DIVIDE WIDTH"
.Cells(31, 2).Value = Centre_Width
.Cells(31, 3).Value = "metres"
End If
```

```
.Cells(33, 1).Value = "LUMINAIRE NAME"
.Cells(33, 2).Value = Worksheets("Photometry").Cells(13, 6).Value
.Cells(34, 1).Value = "LAMP"
.Cells(34, 2).Value = Worksheets("Photometry").Cells(32, 6).Value
.Cells(35, 1).Value = "LAMP LUMENS"
.Cells(35, 2).Value = lamp_lumens
.Cells(35, 3).Value = "lumens"
.Cells(36, 1).Value = "NUMBER OF LAMPS"
.Cells(36, 2).Value = number_of_lamps
.Cells(38, 1).Value = "MAINTAINENCE FACTOR"
.Cells(38, 2).Value = maintainance_factor
'-------------------------------Statistics-----------------------------------------------
.Cells(40, 1).Value = "SCHEME STATISTICS"
.Cells(42, 1).Value = "AVERAGE LUMINANCE"
.Cells(42, 3).Value = "cdm2"
.Cells(42, 2).Value = Ave_Lum
.Cells(44, 1).Value = "OVERALL UNIFORMITY (Uo)"
.Cells(44, 2).Value = Uo_Lum
.Cells(45, 1).Value = "LONGITUDINAL UNIFORMITY (Ul)"
.Cells(45, 2).Value = Ul
.Cells(47, 1).Value = "THESHOLD INCREMENT (TI)"
.Cells(47, 2).Value = (65 / .Cells(42, 2).Value ^ 0.8) * Lv_Total
.Cells(47, 3).Value = "%"
.Cells(49, 1).Value = "Surround Ratio (SR)"
.Cells(49, 2).Value = Worksheets("Results Details").Cells(164 + sheet_loc, 2).Value / Worksheets("Res
sheet_loc, 2).Value
'-------------------------------Grid-----------------------------------------------
'For i = 1 To No_Of_Obs
```

```
'sheet_loc = 20 * (i - 1)
'.Cells(61 + sheet_loc, 1).Value = "Point Locations"
'For j = 1 To points_across
'.Cells(j + 61 + sheet_loc, 1).Value = widepoints(j) 'writes to sheet
'Next j
'For j = 1 To points_long
'.Cells(61 + sheet_loc, j + 1).Value = longpoints(j)
'Next j
'.Cells(60 + sheet_loc, 1).Value = "LUMINANCE VALUES FOR OBSERVER " & i
'l = 62 + sheet_loc
'k = 2
'Do Until .Cells(61 + sheet_loc, k) = ""
'Do Until .Cells(l, 1) = ""
'.Cells(l, k).Value = Worksheets("Results Details").Cells(l + 6, k + 1).Value
'l = l + 1
'Loop
'k = k + 1
'l = 62 + sheet_loc
'Loop
'Next i
.Select
.Cells(1, 1).Select
End With
End Sub
```